PROTOTYPE CONSTRUCTION OF A COMPILER FOR NETWORK ANALYSIS FEASIBILITY STUDY

UCLA-ENG-7041
September 1970

D. F. Martin
L. P. McNamee
R. Chen
D. Patterson

UCLA-ENG-7041
September 1970

PROTOTYPE CONSTRUCTION OF A COMPILER FOR NETWORK ANALYSIS FEASIBILITY STUDY

by

David F. Martin
Lawrence P. McNamee
Richard Chen
David Patterson

School of Engineering and Applied Science
University of California
Los Angeles, California

PREFACE

The research described in this report, "Prototype Construction of a Compiler for Network Analysis Feasibility Study," was carried out under the direction of L. P. McNamee and D. F. Martin, Principal Investigators, and W. J. Karplus and M. A. Melkanoff, Co-Principal Investigators, in the School of Engineering and Applied Science, University of California, Los Angeles.

This project was supported by the Electronics Research Center, National Aeronautics and Space Administration, Contract No. NAS 12-2104.

Other published research supported in part by this contract is reported in Chan, R., et al., "A Programming Language and System for Computer Aided Circuit Analysis/Design," Midwest Circuit Theory Symposium, Minneapolis, Minnesota, May 1970.

Patterson, D. A., "A Programming Language Translation System Based on Generalized Syntax-Directed Transduction," M. S. Thesis, Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, 1970.

Research preformed under NASA Contract No. NAS 12-2105, "Standardization and Qualification of Computer Programs for Circuit Design," directly supported the work reported herein.

CONTENTS

FIGURES

I. INTRODUCTION

The field of computer-aided circuit design has seen the development of many circuit analysis systems and packages implemented in high-level programming languages such as FORTRAN. As is usually the case in such situations, there has tended to be a proliferation of these systems, many of which are variations of each other.

1. Objectives

The main objective of this project was to investigate the feasibility of combining various circuit analysis packages into a single system. This new system would offer the user a single input language, thus eliminating the need for learning a new programming procedure each time a user wanted to use a different circuit analysis package. Advantages in implementation also accrue, in that the best of the common computational packages and functions in many programs can be incorporated into the universal system. Selectable variations of the same functions can also be employed. If in fact the universal system is to be a common "umbrella" under which the best features of various circuit analysis packages are included, then the system must be extendible, to allow for the addition of new packages or features.

2. Scope of Report

At the time of its inception, this project was planned to be of two years' duration, beginning with a one-year contract. During the first year, the closing of NASA Electronic Research Center (from which the contract was awarded) was announced. Even though the second year of the project was abandoned, the results contained in this report do demonstrate that the operation of a universal circuit analysis language and system is feasible. The extensions, planned for the second year of the project, however, were never implemented.

The remainder of this report contains a definition of the circuit design language in Section II, a description of the circuit design system program

organization and implementation in Section III, and a summary and conclusions in Section IV. Program listings are contained in Appendix A and example problems in Appendix B.

II. LANGUAGE DEFINITION

The circuit design language (CDL) will be defined in the spirit, although not the letter, of the Revised Report on the programming language ALGOL 60. Syntactic definition will employ the well-known Backus-Naur form (BNF), and the semantic definition will be given in plain English.

1. Syntax Notation

In describing the syntax of the circuit design language we shall adopt Backus-Naur form (BNF) which employs the following set of syntactic notations:

A. Nonterminal symbols: A combination of lower-case letters, decimal digits and special symbols are used as nonterminal symbols. A nonterminal symbol is embraced in a pair of angle brackets, <, > , e.g., <identifier>, etc.

B. Terminal symbols: A combination of upper-case letters, decimal digits and special symbols, with the first character a letter, are used as terminal symbols. A terminal symbol may read ' FREQUENCY' , ' END PROGRAM' , etc.

C. The symbol ' :: =' separates the left and alternative right hand parts of each production, which means that the symbol on the left hand part can be replaced by one of the alternative sequences of symbols given on the right hand part by the application of the production.

D. Alternatives: Each production may have a number of alternative right hand parts. A vertical bar ' |' , read "or", is used to separate these alternatives. For instance, <component description>:: = <passive element> |
< active element > |
< source element >

2. Coding a Network

A network is a directed graph, which consists of a set of vertices called nodes, and a set of arcs (directed line segments) connecting these nodes. The direction of each arc defines a unique direction of current flow between its terminal nodes. An arc is a branch of the graph and it contains a component. A group of nodes may be connected together by components to form a subnetwork, a block, an n-port or n-terminal network. Each subnetwork, block, n-port, or n-terminal has a name. By combining subnetworks, blocks, n-port and n-terminal networks, we can construct a network.

To program a circuit analysis problem, some preparatory work has to be done:

A. Labeling and coding the network;
B. Writing the input source program in the circuit design language.

We shall now discuss labeling and coding the network. The definition of the circuit design language will be given in Section II. 3.

The coding system is as follows:

A. Each node has a unique index associated with it. Index 0 is assigned to the ground node. The rest of the nodes are indexed 1, 2, 3..., etc., consecutively. A square box contains the node number. An arrow is drawn from the higher numbered towards the lower numbered node which indicates the arbitrarily chosen direction of current flow.

B. A name- identifier is assigned to the network and each subnetwork, block, n-port, or n-terminal network.

C. Each component or device has a name. A series of indices is associated with each type of component or device as well as subnetwork. The component indices are chosen consecutively from 1, 2, 3, ... A character string attached in front of the index is used to denote the type of component, device, or network, such as R-1, R-2 ..., etc. for resistors; L-1, L-2, . . . for inductors; and net-1, net-2, . . . for subnetworks.

The following is a list of component and device identifiers.

Component/device name	Identifier
Resistor	R
Capacitor	C
Inductor	L
Mutual Inductor (coupling coefficient)	K
Impedance	Z
Admittance	Y
Voltage	V
Current	I
Pulse Voltage	PV
Pulse Current	PI
Sine Voltage	SV
Diode	D
Transistor	T
Field Effect Transistor	FET
Special Device	SD
Integrated Circuit	IC
Network/subnetwork	NET
n-port	NPORT
n-terminal	NTER

Example 2.1

Figure 2.1 is a single-stage transistor circuit. It consists of seven nodes, seven components, and one transistor. The coding of this network is shown in Figure II.1.

3. Circuit Design Language Definition

3.1 Program Structure

Syntax

<program> :: = <Sub-program> END PROGRAM |
<Sub-program> <program>

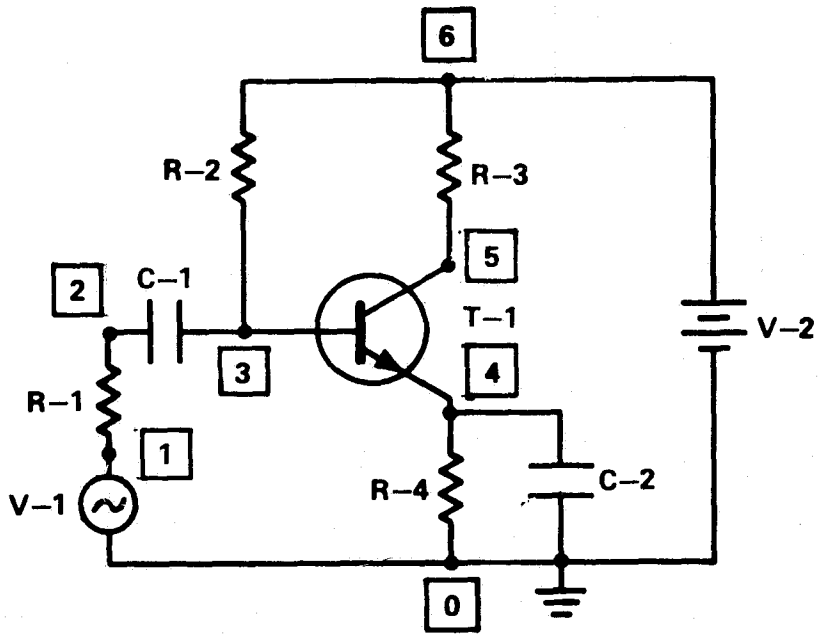


Figure II.1 Circuit for Example 2.1

<Sub-program> :: = <program id> <program test>

END <program id>

<program text> :: = <network description> <command statement> |

<external models > <program text >

Semantics

A circuit design program may consist of one or more concatenated sub-programs. Each subprogram is an independent segment of input language program text describing the network to be analyzed and kind of analysis desired. Each subprogram has an identifier and a network associated with it.

A subprogram consists of a program identifier, an external model set-up if required, a network description, a group of command statements, and finally, a completion statement to indicate the exit point of the subprogram. We shall discuss each section in detail.

A. Program identifier: a name assigned to each subprogram

B. External model: each model is a topological description of an active device' s characteristics. It is required if the device is referred to in a network description.

C. Network description: A topological description of the network as well as the specification of the value or value functions of each component, device, and device model parameters. Every active device should have a model, and the model identifier should be the one that appeared in the external model set-up. If it is not an external model, one of the standard models shall be specified; otherwise, it will be assigned by the system automatically.

D. Command statements: Command statements specify the action to be taken in the course of execution. There are five types of command statement:

a. Execution statements: to specify type of analysis.

b. Modification statements: to modify the network configuration and value of components.

c. Termination statements: to terminate or load a subnetwork if some analysis shall be performed on it.

d. Frequency and time interval statements: to specify the operating frequency and time interval of analysis.

e. Output statement: to specify the type of output and output format.

E. Completion statement: the exit point of the subprogram.

A circuit analysis program is written in free format. Blanks, commas, slashes, and parentheses are delimiters. A detail description of syntax and semantics of each part of the program is given in the following paragraphs.

Examples will be given from time to time for the purpose of illustration.

3.2 Network Description

A network description should consist of the following information:

A. A topological description about how each component and devices are connected;

B. A quantitative expression of the composed component; .

C. Specification of the model desired for each active device.

Syntax

< network description > :: = NETWORK DESCRIPTION
< network >
END NETWORK DESCRIPTION

< network > :: = < network identifier >
< subnetwork >
END < network identifier >

< subnetwork > :: = < component description > |
< Block > |
<network identifier> |
< network > |
<subnetwork><network> |

Semantics

Every network has an identifier which serves as the name of the network. A network is itself a subnetwork or a group of subnetworks connected together. Each subnetwork must have a unique identifier. A single component can also be considered as a subnetwork and possess a component identifier. By a recursive definition, a network may consist of several subnetworks, and each subnetwork may again be considered as a network and consist of subnetworks and/or components. Each subnetwork has an END statement to indicate the completion of its description.

Example 3. 1

An example showing how a network may be constructed as a group of components and subnetworks is given in Figure II. 2. The tree structure shows the inclusion relationship between the main network, NET-1, and its subnetworks. NET-1 consists of two components and two subnetworks, NET-2 and NET 3. Each subnetwork itself consists of components and subnetworks. Figure II. 3 gives a pictorial view of the network configuration. The network (NET-2) contained in the dotted box is a subnetwork. The inner dotted box (NET-4) is a sub-network of the outer box. The program text for the network description of NET-1 is as follows:

```
NETWORK DESCRIPTION
      NET-1
         Comp-1
            Comp-2
      NET-2
      NET-3
END NET-1
      NET-2
         Comp-3
         Comp-4
         NET-4
END NET-2
      NET-4
         Comp-5
END NET-4
      NET-3
      Comp-6
      Comp-7
END NET-3

END NETWORK DESCRIPTION
```

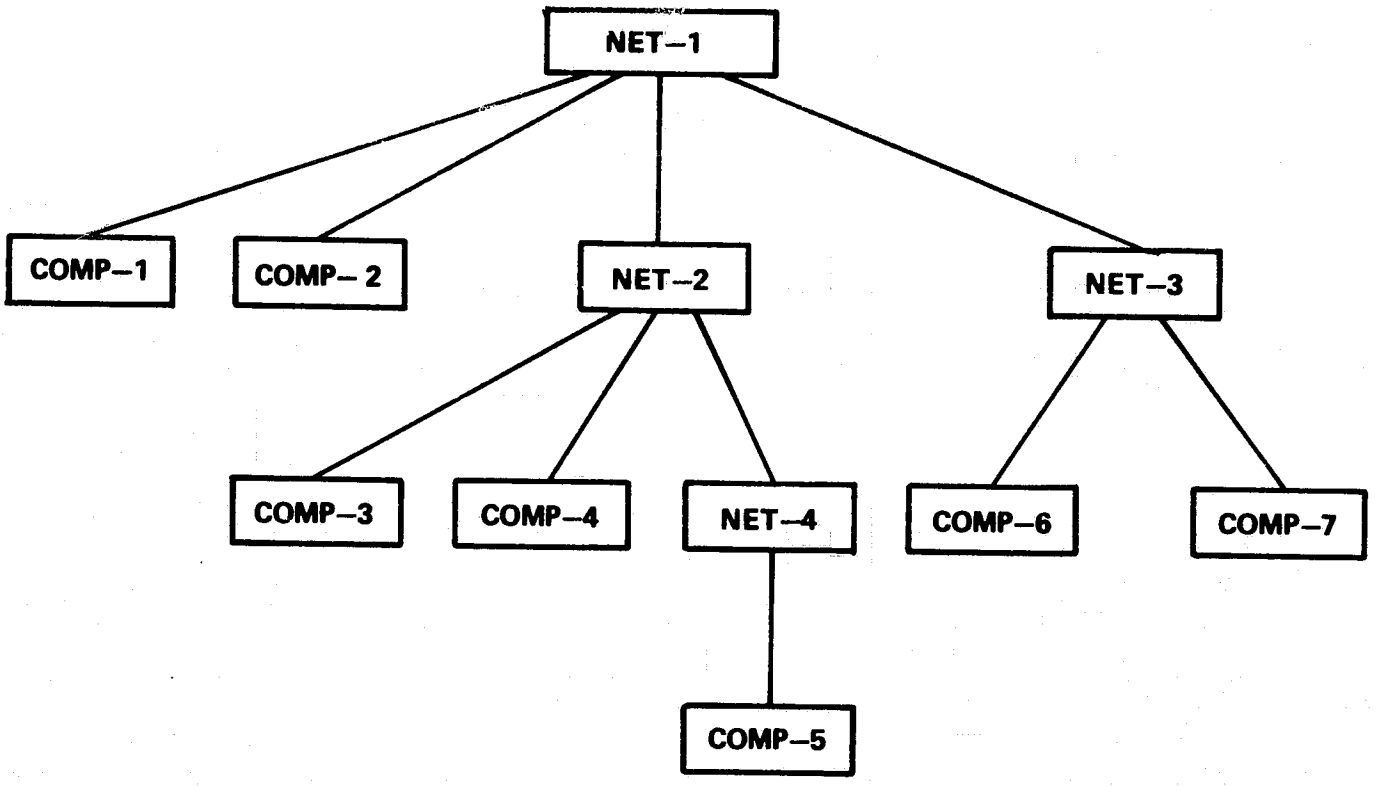


Figure II.3 Tree of Network Description of NET-1

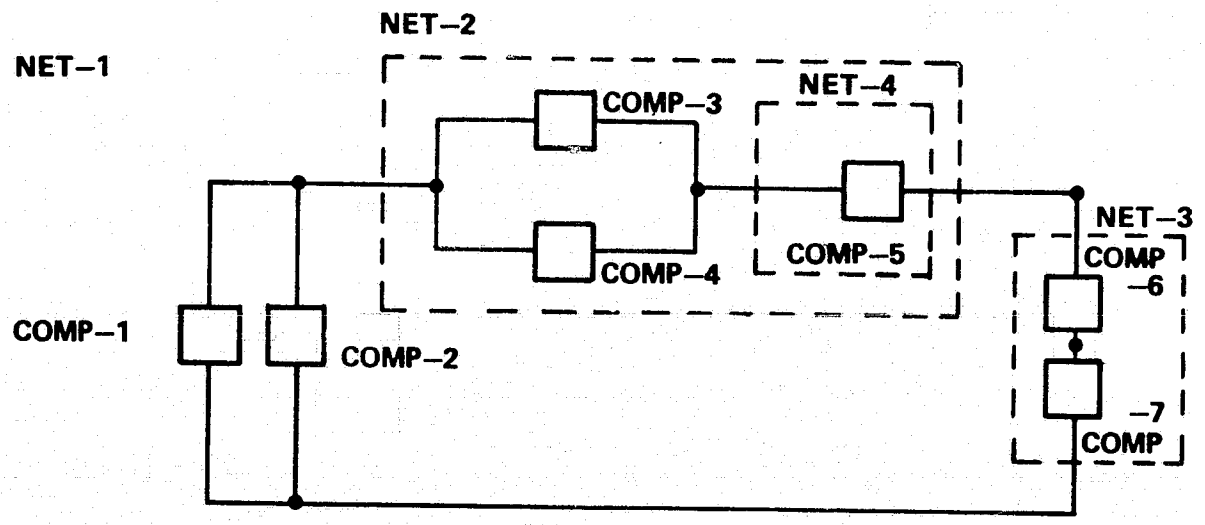


Figure II.3 Network Diagram of NET-1

3.3 Component Description

Passive elements, source elements, and active devices are all considered as basic components of a network. The passive elements and source elements have only two external terminals, whereas the active devices may have more than two external terminals.

The characteristics of active devices change under different operating conditions, and hence it is up to the user to specify a suitable model for each device.

The following are the syntax rules showing how a component description is made up.

Syntax

< Component description > :: = < passive element > |
< active device > |
< source device >

< Passive element > : = < passive element terminals > < resistor > |
< passive element terminals > < capacitor > |
< passive element terminals > < inductor > |
< passive element terminals > <mutual inductor> |
< passive element terminals > < impedance > |
< passive element terminals > <admittance> |

< Source element > :: = < source element terminals > < voltage > |
< source element terminals > < current> |
< source element terminals > <pulse voltage> |
< source element terminals > <pulse current> |
< source element terminals > <sine voltage>

< passive element terminals > :: = NODE (< integer >, < integer >)

< source element terminals > :: = NODE (< integer >, < integer >)

< resistor > :: = < r-id > < value function >

< capacitor > :: < c-id > < value function >

< inductor > :: = < l-id > < value function >

< mutual inductor > :: =< K-id > < value function >

< impedance > :: = < Z-id > < value function >

< admittance > :: = < Y-id > < value function >

< voltage > :: = < V-id > < value function >

< current > :: = < I-id > < value function >

< pulse voltage > :: = < PV-id > < value function >

< pulse current > :: = < PI-id > < value function >

< sine voltage > :: = < SV-id > < value function >

< R-id > :: = R- < integer >

< C-id > :: = C- < integer >

< L-id > :: = L- < integer >

< K-id > :: = K- < integer >

< Z-id > :: = Z- < integer >

< Y-id > :: = Y- < integer >

< V-id > :: = V- < integer >

< I-id > :: = I- < integer >

< PV-id > :: = PV- < integer >

< PI-id > :: = PI- < integer >

< SV-id > :: = SV- < integer >

< component id > :: = < R-id > |
< C-id > |
< L-id > |
< K-id > |
< Z-id > |
< Y-id > |
< V-id > |
< I-id > |
< PV-id > |
< PI-id > |
< SV-id > |
< D-id > |

< T-id > |

< IC-id > |

< SPEC-id >

< value function > :: = < constant > |

< simple arithmetic expression >

< id > : = < program id > |

< network id > |

< subnetwork id > |

< component id >

Semantics

We have considered a network as a directed graph. A unique direction of current is assigned to each component in the network. The convention of current flow is from the higher numbered node (origin) to the lower numbered node (destination). The component description follows this orientation which assigns a direction from n1 (higher number node) to n2 (lower number node).

The value of each component may be a constant or a simple arithmetic expression. The MKS unit system is used for component values. If the value of a component is expressed by an expression, the variables appearing in the expression shall be defined at run time.

For a pulse component element, a special format is used to specify its rise time, fall time, duration, and period. Figures II. 4 and II. 5 show pulse waveforms and how each time interval should be assigned to specify pulses with a variety of shapes.

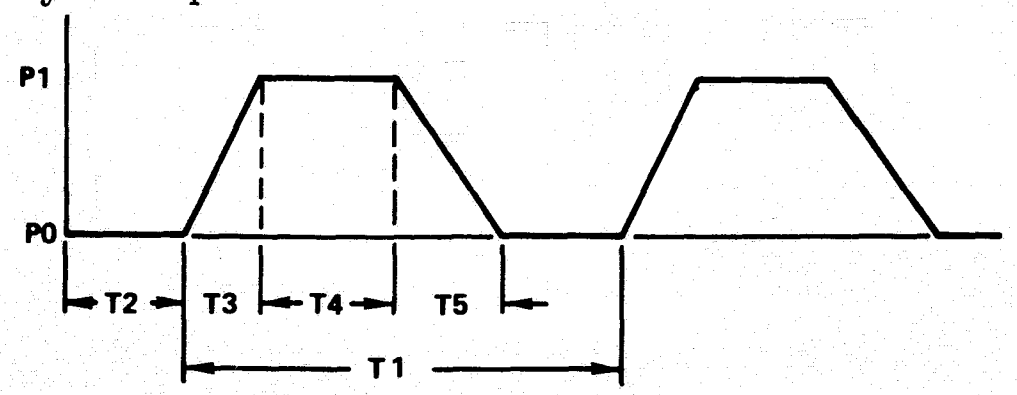
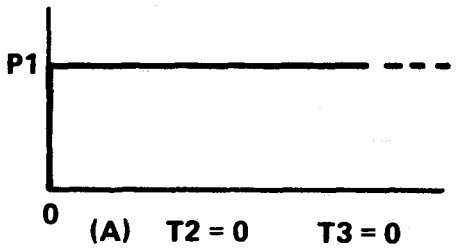
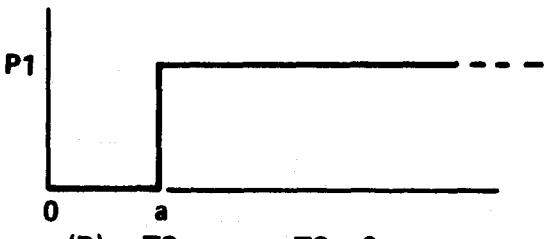


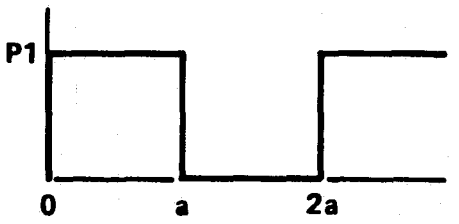
Figure II.4 Pulse Wave Form



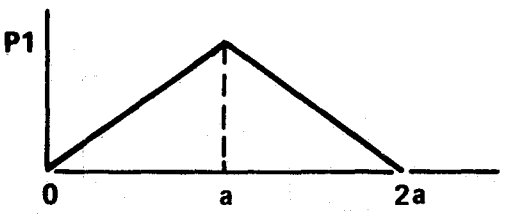
(A) T2 = 0 T3 = 0
T4 = T1 T5 = 0



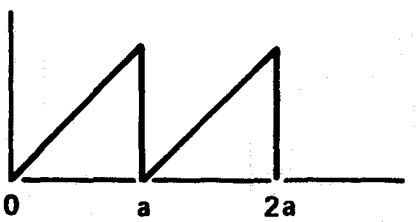
(B) T2 = a T3 = 0
T4 = T1 T5 = 0



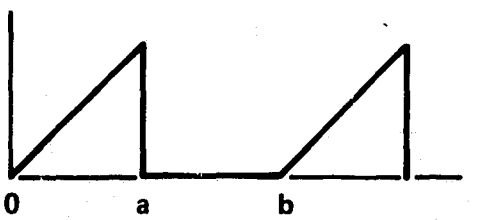
(C) T1 = 2a T2 = 0
T3 = 0 T4 = a T5 = 0



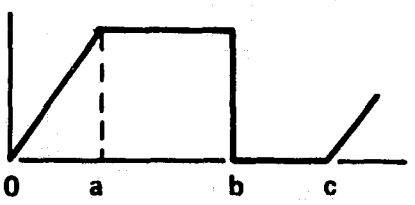
(D) T1 = 2a T2 = 0 T3 = a
T4 = 0 T5 = a



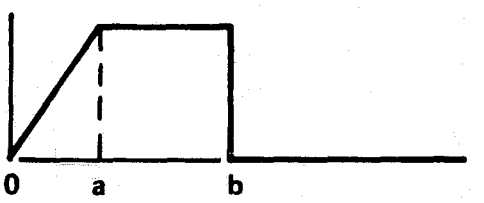
(E) T1 = a T2 = 0 T3 = a
T4 = 0 T5 = 0



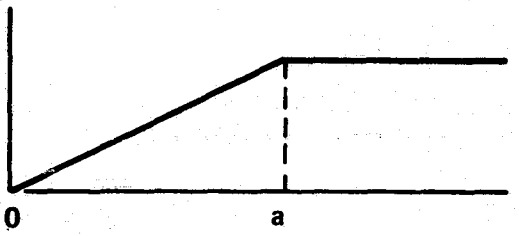
(F) T1 = b T2 = 0 T3 = a
T4 = 0 T5 = 0



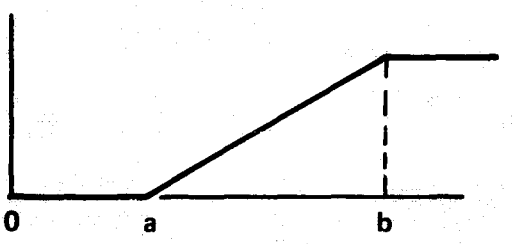
(G) T1 = c T2 = 0 T3 = a
T4 = (b—a) T5 = 0



(H) T2 = 0 T3 = a
T4 = (b—a) T5 = 0



(I) T2 = 0 T3 = a
T4 = T1—a T5 = 0



(J) T2 = a T3 = b—a
T4 = T1—b T5 = 0

Figure II.5 Special Pulses

3.4 Active Devices

Active devices are diodes, transistors, integrated circuits, and other special devices. The following syntax rules describe their external connections, model, and model parameter list.

Syntax

< active device > :: =

< diode terminal > < diode > |

< transistor terminal > < transistor > |

< I. C. terminal > < I. C. > |

< special device terminal > < special device >

< diode terminal > :: = NODE (< integer >, < integer >)

< transistor terminal > :: = NODE (< integer >, < integer >, < integer >)

< IC terminal > :: = NODE (< integer list >)

< special device terminal > :: = NODE (< integer list >)

< diode > :: = < D-id > < model specification >

< parameter value list >

< transistor > :: = < T-id > < model specification >

< parameter value list >

< I. C. > :: = < IC-id > < model specification >

< parameter value list >

< special device > :: = < special device-id > < model specification >

< parameter value list >

< D-id > :: = D- < integer >

< T-id > :: = T- < integer >

< IC-id > :: = IC- < integer >

< special device-id > :: = SPEC- < integer >

< model specification > :: = STANDARD < id >

EXTERNAL < id >

< parameter value list > :: =

< passive element id > = < value function > |

< source element id > = < value function > |

< parameter value list > < parameter value list >

< id > :: = < letter > |

< id > < letter > |

< id > < digit >

Semantics

A. A diode has two terminals, a transistor has three terminals, and a special device or an integrated circuit has three or more external terminals.

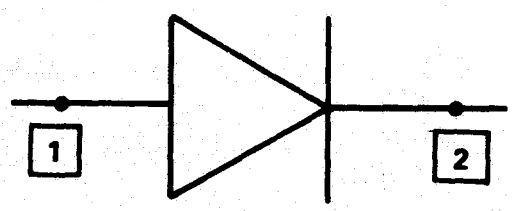
B. For diode terminals NODE(n1, n2), n1 is the anode, and n2 the cathode. For transistor terminals NODE(n1, n2, n3), n1 is the emitter, n2 the base, and n3 the collector. As for a special device or an integrated circuit, the numbering of the terminal nodes may be listed in a specific order.

C. Device models may be either standard or external. A list of standard model names built into the system is given in the Appendix. If the standard model is not used, an external model shall be specified and the model name shall be one of the external model identifiers defined in the program, and it shall be different from any of the standard model identifiers.

D. The parameter value list provides the value or expression for each component in the model. The model is treated as a subnetwork embedded in the main network.

The following are examples showing how the active devices are set up.

Example 3.2 Diode D-1



D–1

Figure II.6 Diode

Example 3.3 Transistor T-1

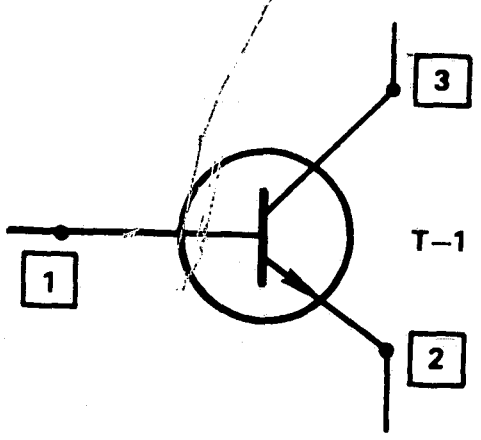


Figure II.7 Transistor

3.5 Blocks

Syntax

< block > :: = < n-port > |
< n-terminal >

< n-port> :: = < n-port id > < n-port list >
< n-port configuration >

< n-port list > :: = PORT (n1: n2 : n3 : n4,...)

< n-terminal list > :: = TERM (n1, n2,...)

< n-port id > :: = NP- < integer >

< n-terminal id > :: = NT- < integer >

< n-port configuration > :: = < subnet description >

< n-terminal configuration > :: = < subnet description >

Semantics

A block is a subnetwork which has n external ports or n external terminals and is called an n-port or n-terminal network. The most frequently used are two port and two terminal networks. Often some of the ports or terminals are considered as input ports where others are output, control or sensing ports or terminals. Each port of a block is described as a pair of nodes separated by a semicolon, ports being separated by commas. A configuration

of the block shall be followed in order to describe the internal composition of the n-port/n-terminal network.

Example 3.4

An example of a n-port and n-terminal network description is given here. The circuit diagrams are given in Figures II.8 and II.9, respectively.

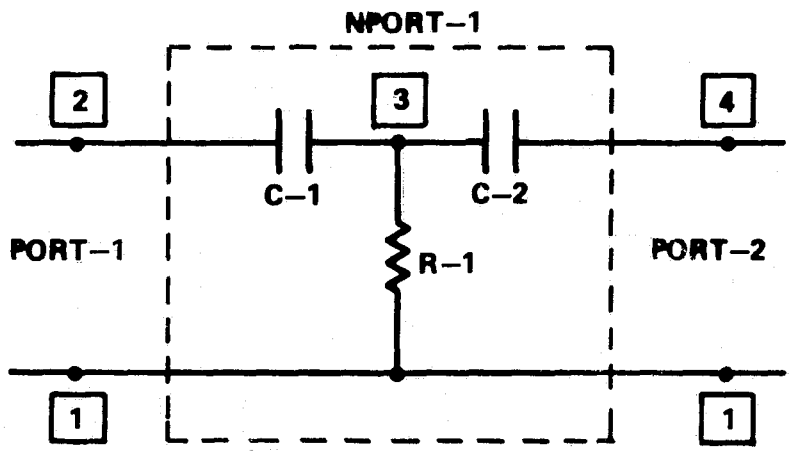


Figure II.8 Circuit Diagram for NPORT-1

Network description for NPORT-1:

NPORT-1

PORT (2:1, 4:1)

NODE (3, 2) C-1 = 0.01

NODE (4, 3) C-2 = 0.01

NODE (3, 1) R-1 = 1000

END NPORT-1

Network description for a n-terminal network:

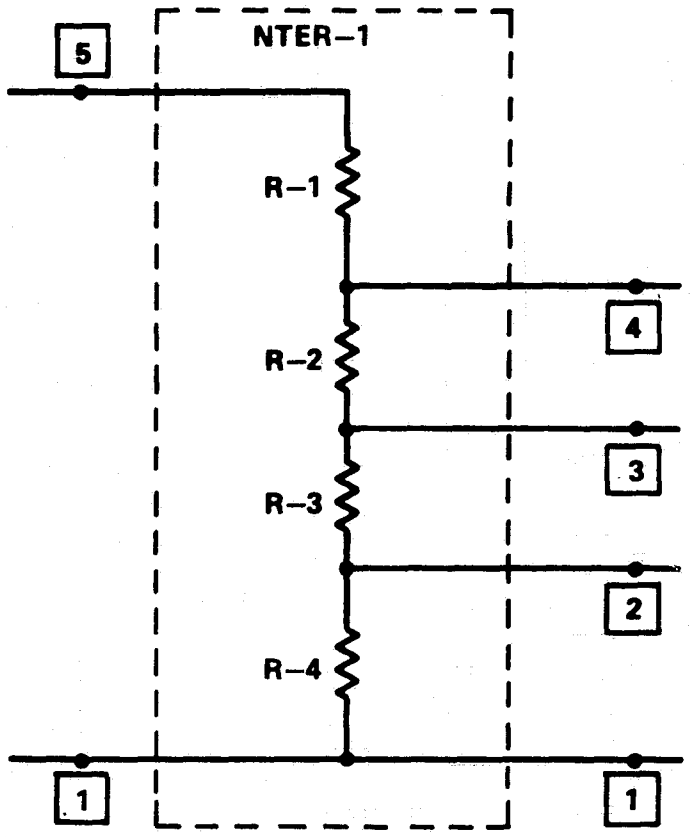


Figure II.9 Circuit Diagram for a n-Terminal Network

```
NTER-1
    TERM (1, 2, 3, 4, 5)
    NODE (5, 4) R-1 = 2500
    NODE (4, 3) R-2 = 2500
    NODE (3, 2) R-3 = 2500
    NODE (2, 1) R-4 = 2500
END NTER-1
```

3.6 External Models

Syntax

< external model > :: = EXTERNAL MODEL
< model >
END EXTERNAL MODEL

< model > :: = < model id > < terminal nodes list >

<model set up > END < model id > |

< model > < model >

< model set up > :: = NODE (n1, n2)

< parameter id > |

< model set up > < model set up >

< parameter id > :: = < passive element id > |

< source element id >

< terminal nodes list > :: = (< integer list >)

Semantics

An external model is a topological description of the characteristics of an active device. The external model is required whenever ' EXTERNAL MODEL' is specified in the network description for that device. A name (or identifier) is assigned to each model. The terminal node list of the model must agree with the device external terminal nodes in number and order. For instance, the order of the terminal nodes for a transistor is emitter, base, collector. Several different models may be cascaded. Whenever a model is called for, its configuration is effectively substituted into the corresponding device. For a particular model set-up, it can be called at several places in the network description. An active device can have several different models if it used in several places in a network.

Example 3.5

A hybrid - II model is described for a transistor' s equivalent circuit:

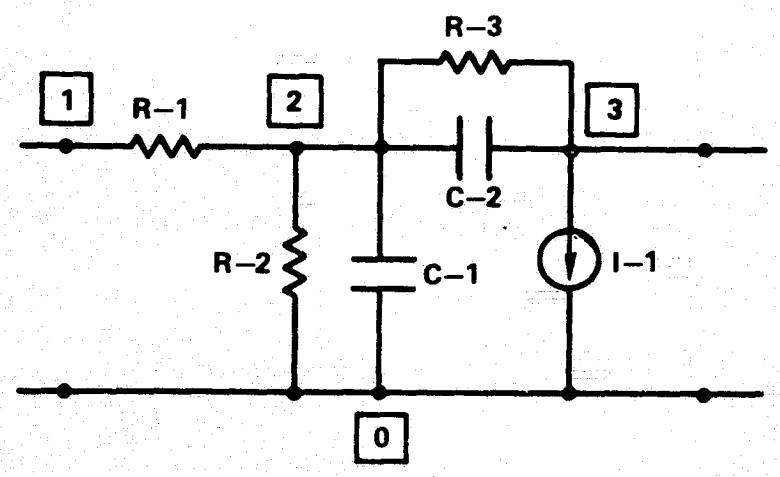


Figure II.10 Hybrid–π Model (HYPI 1)

```
EXTERNAL MODEL
    HYPI1
    (0, 1, 3)
    NODE (2, 1) R-1
    NODE (2, 0) R-2
    NODE (2, 0) C-1
    NODE (3, 2) R-3
    NODE (3, 0) I-1
    END HYPI1
END EXTERNAL MODEL
```

3.6 Command Statements

The network description and external model provide information only about the inter-connection among the components of the network and the value or value expression of each component. The command statements to be discussed next provide a step-by-step execution outline for the circuit analysis, network modification, and outputting the analysis result.

Syntax

```
< Command statement > :: = < execution statement                    |
                            < modification statement >               |
                            < termination statement >                |
                          < frequency and time interval statement >  |
                            < output statement >
```

3.6.1 Execution Statement

Syntax

```
< execution statement > :: = EXECUTE < network id >
                                     < type of analysis >
                                     < routine id >
< type of analysis > :: = DC | AC | TRANSIENT
```

< routine id > :: = < id >

< network id > :: = < id >

Semantics

There are DC, AC, and Transient analyses for a network. Other analysis may also be defined. Each execution statement must specify which network is analyzed. Special routines may be chosen by the programmer for certain circuit analyses in order to take full advantage of their power. For example, routine A may be good for AC analysis at low frequencies, whereas routine B might be good for transient analysis.

An execution routine is referred to by name, and it exists in the system in order to be available for call.

Example 3.6

The execution statement

EXECUTE NET-1 AC ECAP

means that AC analysis is to be performed on network NET-1 by routine ECAP.

Example 3.7

The execution statement

EXECUTE NET-2 TRANSIENT CIRCUT

means transient analysis is to be performed on network NET-2 by routine CIRCUS.

3.6.2 Modification Statements

Syntax

< modification statement > :: = < type of modification >
< modifying component >

< type of modification > :: = MODIFY | DELETE | ADD

< modifying component > :: = NODE (< integer >, < integer >)
< component id > |

NODE (< integer >, < integer >)
< component id >
= < new value >

< new value > :: = < constant > | < simple arithmetic expression >

Semantics

Often the network topology or network component values have to be changed during the process of design. Especially for designs by trial-and-error methods, modification of network elements is expected. The modification statements allow the programmer to perform analysis and then modify his network according to the result.

There are three different types of modifications. MODIFY is used to change a component value and DELETE is used to delete a component from a network. ADD is used to connect a new component between two nodes of the network under analysis. Each one of the modification statements can only affect one component description or one device specification, and hence a combination of the three types of modification statements is required to change a device model. If this is done, a new parameter value list must be made in order that the new model be valid.

Example 3.8

```
MODIFY NODE (1, 2)  R-1=300
DELETE NODE (5, 4)  C-2
ADD NODE    (5, 4)  R-5 = 3000
```

The first statement above changes the value of R-1 to a new value of 300 ohms. The second statement will delete C-2 from the network, and then a resistor R-5 of 3000 ohms will be added to the circuit in place of C-2.

3.6.3 Termination Statements

Syntax

< termination statement > :: = < short circuit statement > |
< open circuit statement > |
< termination statement >

< short circuit statement > :: = SHORT < network id >
NODE (< integer >, < integer >)

< open circuit statement > :: = OPEN < network id >
NODE (< integer >, < integer >)

< termination statement > :: = TERMINATE < network id >
<component description>

Semantics

In engineering problems, one often expects to conduct short circuit and open circuit tests on a circuit to obtain some special test measurements or to do circuit diagnosis. The short and open circuit statements are designed for this purpose. When ' OPEN' or ' SHORT' is applied to any pair of nodes, all the elements connecting such two nodes are deleted or short circuited tentatively for the corresponding analysis. The termination statement has a special use. Since we are allowed to conduct any defined kind of analysis on a particular subnetwork, termination of its terminals after a subnetwork has been isolated from the main network must be considered. In addition to connected loads, one may have to supply sources in order to perform the analysis.

Example 3.9

```
SHORT NET-1  NODE (3, 5)
OPEN  NET-2  NODE (7, 9)
TERMINATE  NET-3  NODE (2, 1)  R-10=100
```

3.6.4 Frequency and Time Interval Statements

Syntax

<frequency statement> :: = FREQUENCY <lower frequency bound>,
<frequency increment>, <upper frequency bound> |
FREQUENCY <lower frequency bound> /
<frequency multiplicity> / <upper frequency bound>
<time interval statement> :: = TIME <lower time bound>,
<time increment>, < upper time bound> |
TIME <lower time bound> /
<time multiplicity > / <upper time bound>
<lower frequency bound> :: = <constant >
<upper frequency bound> :: = <constant >
<frequency increment> :: = < constant>
<frequency multiplicity> :: = < constant>
<lower time bound > :: = < constant>
<upper time bound > :: = < constant>
< time increment > :: = < constant>
< time multiplicity > :: = < constant>

Semantics

The frequency statement is used to describe some conditions of AC analysis. It specifies the frequency bound for an analysis and the increment or multiplicity of the frequency interval during the execution of the analysis.

The time interval statement is used for the specification of time parameters of transient analysis. The analysis time bounds and the time interval for each execution analysis segment are specified.

< variable > :: = V - < integer > | I - < integer >
< independent variable > :: = TIME | FREQUENCY
< scale > :: = (< lower bound >, < increment >,

< upper bound >) |

(< lower bound > | < multiplicity >

< upper bound >)

< Range :: = (< lower plot bound > , < plot interval > ,

< upper plot bound >)

(< lower plot bound > | < plot multiplicity >

| < upper plot bound >)

< lower bound > :: = < number >

< upper bound > :: = < number >

< increment > :: = < number >

< multiplicity > :: = < number >

< lower plot bound > :: = < number >

< upper plot bound > :: = < number >

< plot interval > :: = < number >

< plot multiplicity > :: = < number >

Semantics

There are two ways to output results. They may be printed or plotted. If print output is required, one may expect to obtain the output for all current and voltage variables by an implicit print statement. A statement ' PRINT CURRENT' will output all the branch currents of the network or subnetwork just analyzed. If the statement is ' PRINT CURRENT VOLTAGE' , the output consists of two parts, the first being branch currents and the second node voltages. A specific branch current or voltage may be output by an explicit print statement either according to the user' s format or a built-in format.

For a plotted output, the independent variable may be time or frequency. Two quantities must be specified, scaling and plot range. Scaling provides the information about the lower and upper limits of the dependent variable as well as its scaling. Plot range specifies the lower bound, upper bound, and plot interval of the independent variable. Each scale can be either linear or logarithmic.

The lower bound and upper bound of frequency or time variables give the exact range over which an analysis is to be performed. The increment indicates the values between two analysis points, and multiplicity provides a logarithmic incrementation of frequency or time variable.

3.6.5 Output Statements

Syntax

< output statement > :: = < print statement > |
< plot statement >

< print statement > :: = < implicit print > |
< explicit print >

< implicit print > :: = PRINT CURRENT |
PRINT VOLTAGE |
PRINT CURRENT, VOLTAGE

< explicit print > : : = PRINT < variable list > |
PRINT (format label >) < variable list >
< format statement >

< variable list > :: = V - < integer > | I - < integer > |
V - < integer > < variable list > |
I - < integer > < variable list >

< format label > :: = < label >

< label > :: = < integer >

< format statement > :: = < legal FORTRAN format statement >

< plot statement > :: = < PLOT < variable >
< independent variable >
< scale > < range >

3.7 Ending Statement

A program is concluded by an ending statement which consists of an ' END' and the program identifier. The ending statement serves the purpose of bracketing a group of statements as an independent program. An ending statement may also be used to separate two sections of a program or a

subprogram. It functions like a closing parenthesis, and hence for each ending statement, there is a corresponding identifier to which the statement refers.

Syntax

```
< ending statement > :: = END  < id >                    |
                           END   PROGRAM                 |
                           END   NETWORK DESCRIPTION     |
                           END EXTERNAL MODEL

< id > : : =   < program id >   |
               < network id >   |
               < model id >
```

Semantics

END PROGRAM: completion of a program;

END NETWORK DESCRIPTION: completion of a network description;

END EXTERNAL MODEL: completion of the external model set-up;

END <id> : completion of a subprogram, network, subnetwork description, or external model specification.

4. System Active Device Models

In this section the standard (built-in) system models for diodes and transistors will be described.

4.1 Diode Model

A widely used model for the junction diode, the charge-control model, is included in the system as a standard model. The model is shown in Figure II. 11 and its CDL description follows the circuit.

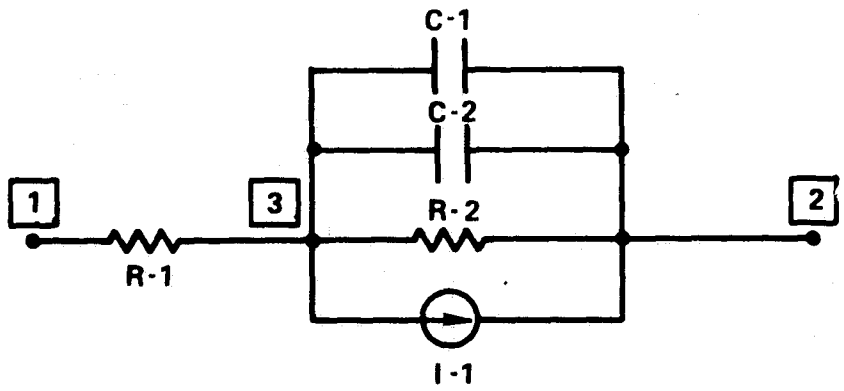


Figure II.11 Charge-Control Model for a Junction Diode

CHARGE CONTROL MODEL

NODE (1, 3) R-1

NODE (3, 2) R-1

NODE (3, 2) I-1

NODE (3, 2) C-1

NODE (3, 2) C-2

MODEL PARAMETER

R-1, R-2, IS, M, K, T, PHI, ND, KT, KD

MODEL EQUATION

I-1 = IS * (EXP(Q*VC1/(M*K*T))-1)

C-1 = KT/((PHI-VCI) ** ND)

C-2 = Q*(I1 + IS) / (6.2832*M*K*T*KD)

END CHARGE CONTROL MODEL

The equations for the charge-control diode model are

$$Id = I_s \left(\exp^{\frac{qV}{MKT}} - 1\right)$$

$$Ct = \frac{K_t}{(\phi - V)^{N_d}}$$

$$Cd = \frac{q(Id + I_s)}{2\pi\ MKT\ Kd}$$

where:

V	------- 'VC1' ,	the junction voltage.
I-1	------- 'I1' ,	diode junction current
C-1	------- 'C1' ,	junction transition capacitance.
C-2	------- 'C2' ,	diffusion capacitance.
R-1	------- 'R1' ,	bulk resistance.
R-2	------- 'R2' ,	chronic leakage resistance.
I_s	------- 'IS' ,	saturation current.
q	------- 'Q' ,	electronic charge.
M	------- 'M' ,	proportional constant
K	------- 'K' ,	Boltzman constant.
T	------- 'T' ,	Absolute junction temperature in ^{o}K.
K_t	------- 'KT' ,	proportional constant.
K_d	------- 'KD' ,	proportional constant.
ϕ	------- 'PHI' ,	junction contact potential, between
	,	0.7 and 1.0V for Si at 25^{o}C.
N_d	------- 'ND' ,	grading constant
		0.5 for abrupt junction
		0.33 for uniformly graded junction

4.2 Transistor Models

A standard system model for a transistor is the Ebers-Moll model. This model is shown in Fig. II.12. The CDL model description is

EBERS-MOLL MODEL

NODE (1, 4) R-1

NODE (2, 5) R-2

NODE (4, 5) R-4

NODE (4, 5) C-1

NODE (4, 5) C-3

NODE (4, 5) I-1

NODE (3, 6) R-3

```
NODE (6, 5) R-5
NODE (6, 5) C-2
NODE (6, 5) C-4
NODE (6, 5) I-2
IES, ICS, BELTAN, BELTAI, ME, MC, K, T, NE, NC, FE, FC,
PHIE, PHIC, AE, AC
MODEL EQUATION
IEF = (IES/(1-ALPHAN*ALPHAI)*(EXP(Q*V1/ME*K*T)-1)
ICF = (ICS/(1-ALPHAN*ALPHAI)*(EXP(Q*V2/MC*K*T)-1)
I1   = IEF = ALPHAI * ICF
I2   = ICF - ALPHAN * IEF
ALPHAN = BELTAN/(1 + BELTAN)
ALPHAI  = BELTAI/(1 + BELTAI)
C1  = AE/(PHIE - V1) ** NE
C2  = AC/(PHIC - V2) ** NC
C3  = Q * (IEF + IES/(1-ALPHAN * ALPHAI))/(2*3.1416*ME*K*T*FE)
C4  = Q * (ICF + ICS/(1-ALPHAN*ALPHAI))/(2*3.1416*MC*K*T*FC)
END EBERS - MOLL MODEL
```

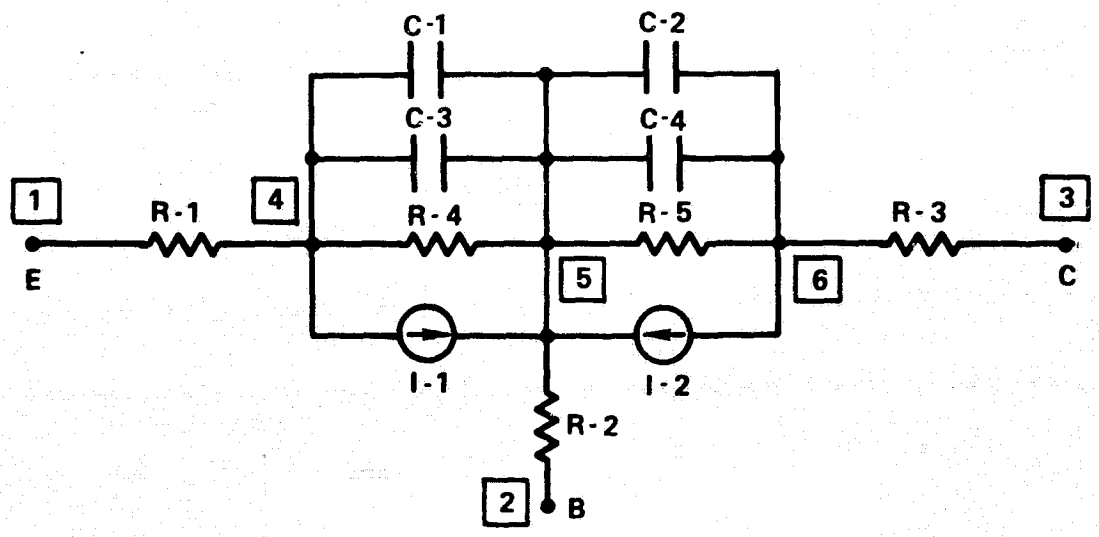


Figure II.12 Ebers-Moll Model for Transistor

Model equations:

label:

$$C_{TE} = \frac{A_e}{(\phi_e - V_1)^{NE}} \qquad \text{C-1}$$

$$C_{TC} = \frac{A_c}{(\phi_c - V_2)^{NC}} \qquad \text{C-2}$$

$$C_{DE} = \frac{q(I_{ef} + \frac{I_{es}}{1-\alpha_n\alpha_i})}{2\pi\, M_c\, K\, T\, F_e} \qquad \text{C-3}$$

$$C_{DC} = \frac{q(I_{cf} + \frac{I_{cs}}{1-\alpha_n\alpha_i})}{2\pi\, M_c\, KT\, F_c} \qquad \text{C-4}$$

$$I_1 = I_{ef} - \alpha_i I_{cf} \qquad \text{I-1}$$

$$I_2 = I_{cf} - \alpha_n I_{ef} \qquad \text{I-2}$$

$$I_{ef} = \frac{I_{es}}{1 - \alpha_n\alpha_i} \left(\exp^{\frac{qV_i}{M_c KT}} - 1\right) \qquad \text{IEF}$$

$$I_{cf} = \frac{I_{cs}}{1 - \alpha_n\alpha_i} \left(\exp^{\frac{qV_2}{M_c KT}} - 1\right) \qquad \text{ICF}$$

$$\alpha_n = \frac{\beta_n}{1 + \beta_n} \qquad \text{ALPHAN}$$

$$\alpha_i = \frac{\beta_i}{1 + \beta_i} \qquad \text{ALPHAI}$$

where:

A_e ---------- 'AE' , proportional constant

A_c ---------- 'AC' , proportional constant

ϕ_e ---------- 'PHIE', emitter-base junction contact potential

ϕ_c ---------- 'PHIC', collector-base junction contact potential

N_e ---------- 'NE' , emitter-base grading constant

N_c ---------- 'NC' , collector-base grading constant

α_n ---------- 'ALPHAN', common base normal current gain

α_i ---------- 'ALPHAI', common base inverted current gain

I_{es} ---------- 'IES' , emitter base Saturation current

I_{cs} ---------- 'ICS' , collector base saturation current

M_e ---------- 'ME' , emission constant for emitter

M_c ---------- 'MC' , emission constant for collector

T ---------- 'T' , junction temperature in ^{o}K

β_n ---------- 'BELTAN' , normal current gain

β_i ---------- 'BELTAI' , inverse current gain

F_e ---------- 'FE' , proportional constant

F_c ---------- 'FC' , proportional constant

and

R-1 ---------- r_{ee} , emitter bulk resistance in ohms.

R-2 ---------- r_{bb} , base spreading resistance in ohms.

R-3 ---------- γ_{cc} , collector bulk resistance in ohms.

R-4 ---------- γ_e , emitter-base junction leakage resistance in ohms.

R-5 ---------- γ_c , collector-base junction leakage resistance in ohms.

The user may define his own transistor model. In this case, an external model set-up is used. The format for setting up the external model was described in Section II. 3.6.

III. PROGRAM ORGANIZATION AND IMPLEMENTATION

1. Introduction

Section III of this report will describe the organization and implementation of the circuit design compiler and associated system. First the general organization of the system is described, and then the various system components are discussed in some detail. Finally, methods for making the system "open-ended" through extensibility are discussed.

2. General System Philosophy and Organization

As indicated in Section I, the general system philosophy is the combination of various circuit analysis packages through a common input language and master control program. The overall system organization is shown in Figures III. 1(a) and (b), and III. 2.

2. 1 Brief System Overview

The heart of the system is the circuit design compiler and the interface programs. The compiler produces a FORTRAN master control program and the transformed data. The master-control program calls the interface programs and passes the transformed data to them. These interface programs call on their circuit analysis programs and output the results specified by the programmer in the circuit design language.

The main portion of the system is the compiler. This compiler is of the pattern recognition type. Upon recognizing certain valid key words (i.e., NETWORK, PRINT), it produces some statements of the FORTRAN master program. Since the language is free format, the first phase of the compiler is lexical analysis, with blanks, commas, dashes and parentheses used as delimiters. Upon completion of compilation, the compiler writes suitably coded data to be passed to the interface programs, and gives control to the master control program. The compiler will detect syntactic errors, whereas run-time errors such as ill-conditioned circuit equations, etc. are detected by the circuit analysis program modules (CAPM' S).

The compiler allows FORTRAN statements to be mixed anywhere within the circuit design language program. Note that these statements are passed directly into the master control program, and are not modified by the compiler. For this inclusion of FORTRAN statements to be useful, the user must understand the interaction of his inserted FORTRAN statements with the FORTRAN statements produced by the circuit design compiler. However, the control program consists mainly of declarations and calls to the interface subroutines.

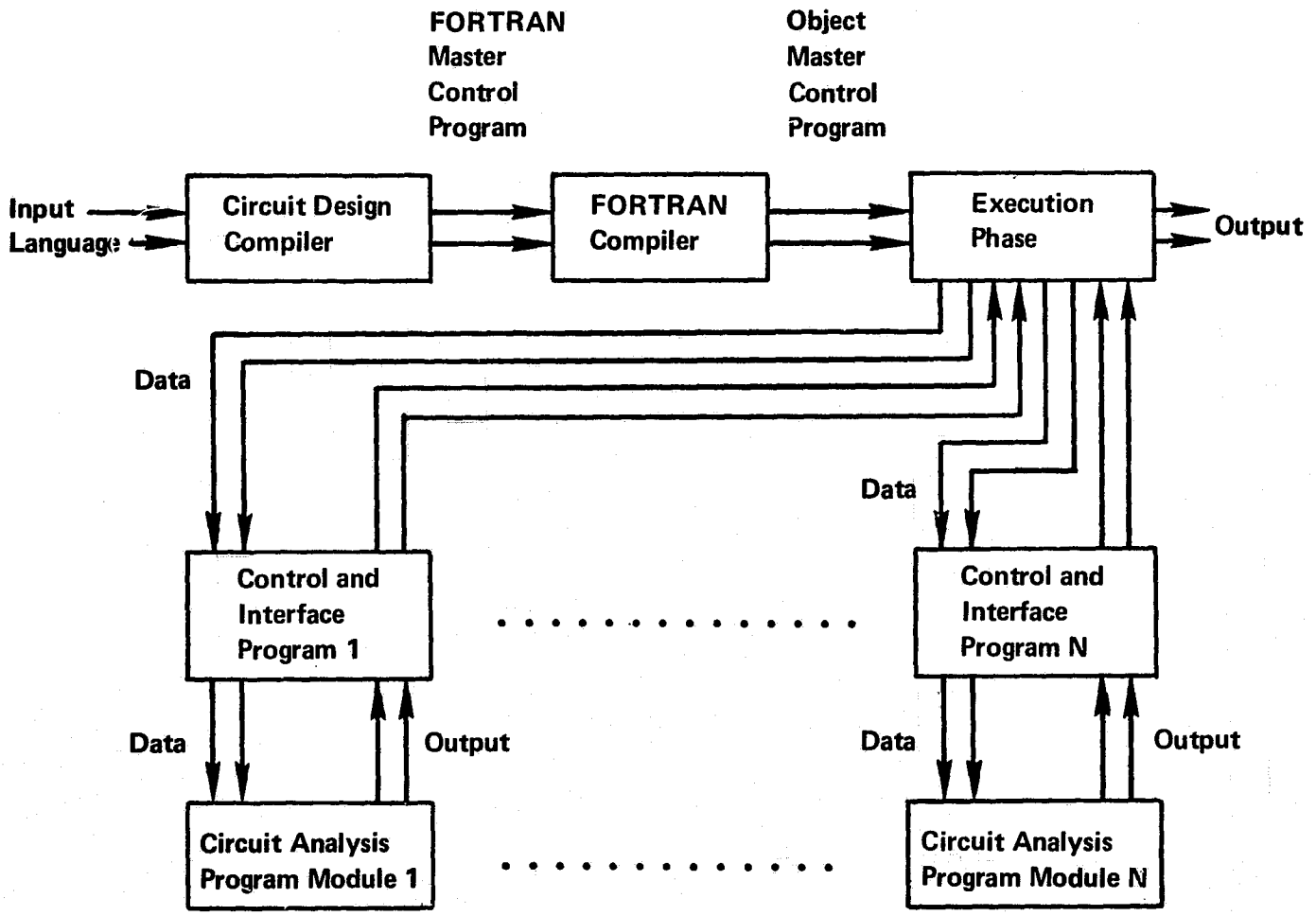


Figure III.1(a) Information Flow Diagram

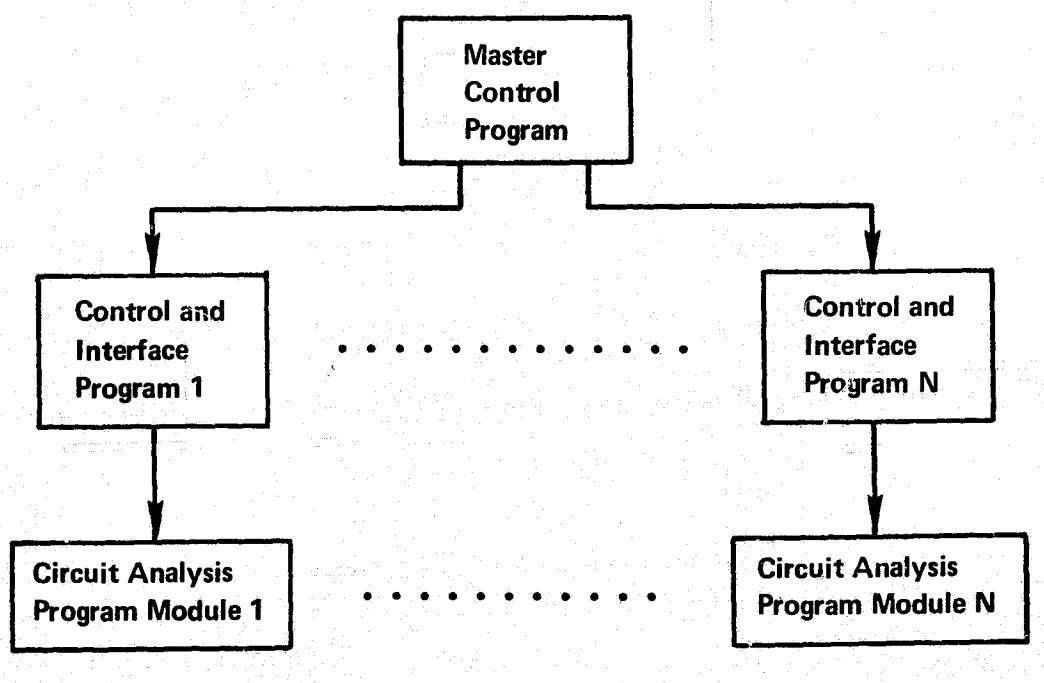


Figure III.1(b) Routine Hierarchy Diagram

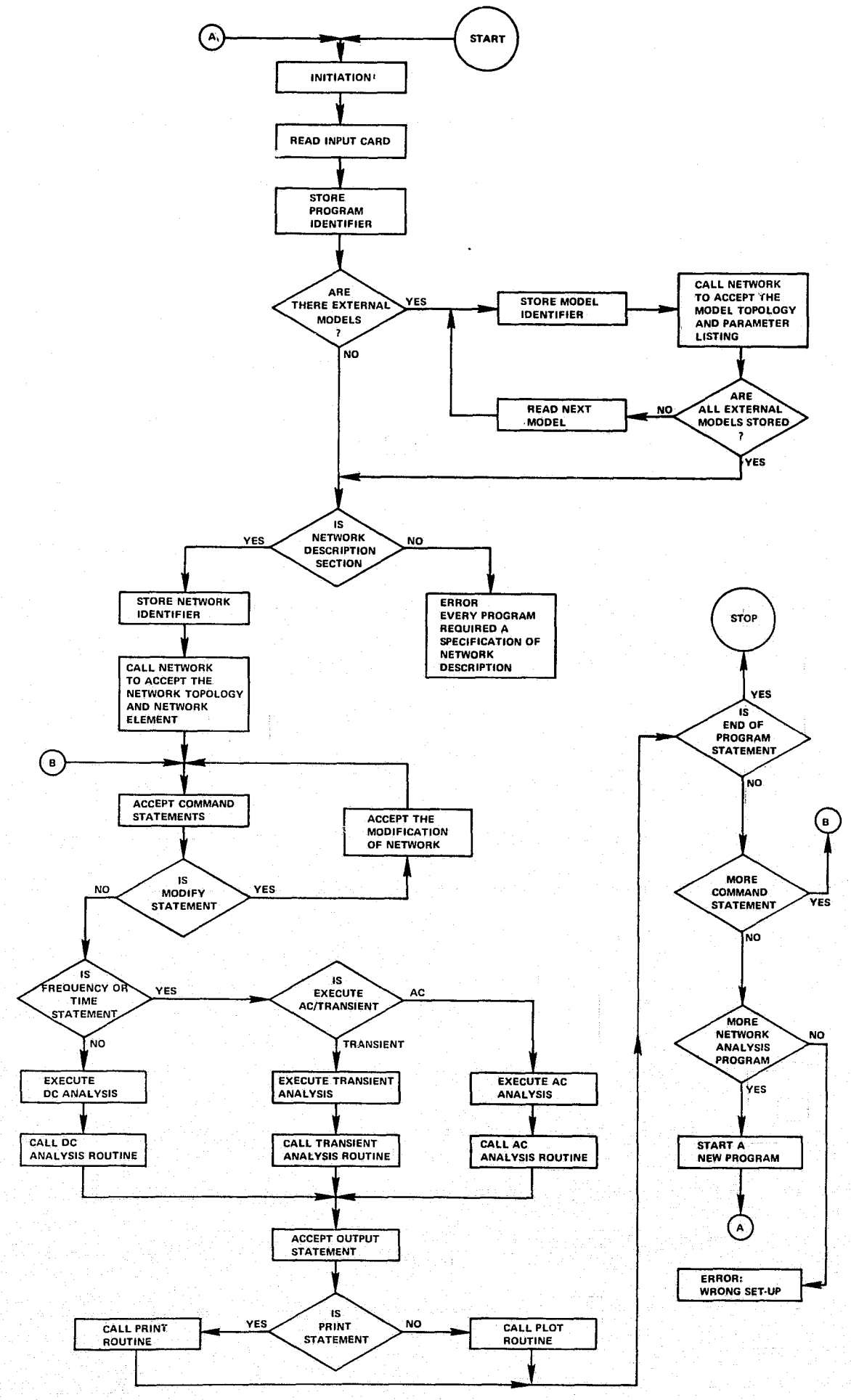


Figure III.2 Flow-Chart for Main Program of Circuit Analysis System

Nearly all the various circuit analysis packages are written in FORTRAN, and nearly all have a structure in which the main program consists only of declarations and call to a few main subroutines. These subroutines then call other subroutines where actual circuit analysis is done . Third, the CAPM' s have five main phases: (a) Input and lexical analysis, (b) Syntax analysis and data validity verification, (c) Storage of the data, (d) Execution, and (e) Output. Often some of these phases are interleaved. For example, a card is read, lexical analysis is performed, the statements on the card are checked to see if they are in the correct form, and they are stored. There is usually a separation between the first three phases, the execution phases, and the output phase.

The circuit design system performs the first two phases in the circuit design compiler, the data storage in the interface subroutines, and execution and possibly output in the CAPM' s. Since a subroutine organization is used, the data can be stored and passed to the subroutines in two ways: through COMMON or through an argument list. A control and interface program then calls all the subroutines that would have been called after the CAPM had finished the first three phases, and then transformed data are stored in the COMMON area or in the variables appearing in the argument lists of the subroutines called. If the particular CAPM requires some execution after data storage, the system merely sets up the storage by establishing a pointer to this data.

3. The Circuit Design Compiler

The major effort required in the implementation of the Circuit Design System was the design and implementation of the Circuit Design Compiler. The principal subtasks of compilation in the system compiler are

(1). Lexical Analysis;

(2). Syntactic Analysis;

(3). Circuit Topology Data Storage.

3.1 Lexical Analysis

Lexical analysis is the subtask whereby circuit design language input, in the form of card images, is processed character by character and grouped and converted into units. The units, called words are in a convenient form for subsequent syntactic analysis. Special characters, call delimiters, aid in the grouping. Blanks, commas, dashes, and parentheses are used as delimiters. The circuit design language is "almost" free-format; card image boundaries are used as statement delimiters.

Lexical analysis is done by the GET, PUT and FETCH subroutines. GET and PUT are subroutines written in assembly language. GET' s arguments are I, J, and K. I is an integer array (of unspecified length), whose elements are characters. GET takes the J^{th} character from I, and places that character, left-adjusted, into K. PUT is the inverse of GET; PUT takes the leftmost character from K and places it in the J^{th} position in the array I.

FETCH does most of the lexical analysis. The flowchart for FETCH is shown in Figure III. 3. FETCH is called by the compiler to break up every card image into separate words. FETCH is also used to break up an array of characters, not just those cards read by FETCH (see NETWORK pulsed elements). If FETCH is not used to break up an array, it reads a card. The first character of each card image is examined to see if the card is a comment card ('C' in first column), or a FORTRAN card to be inserted in the Master Control Program ('*' in the first column). If not, the card image continues to be processed in lexical analysis.

Each group of non-delimiters enclosed by delimiters becomes a new word. If a dash is encountered, it acts as a delimiter, indicating that the next word is to be converted from character to an equivalent integer. An equal sign (=) causes a new word to be formed, causing the rest of the card image to be stored in the value list. If a left parenthesis is encountered, all the following words, up to but not including the next right parenthesis, are

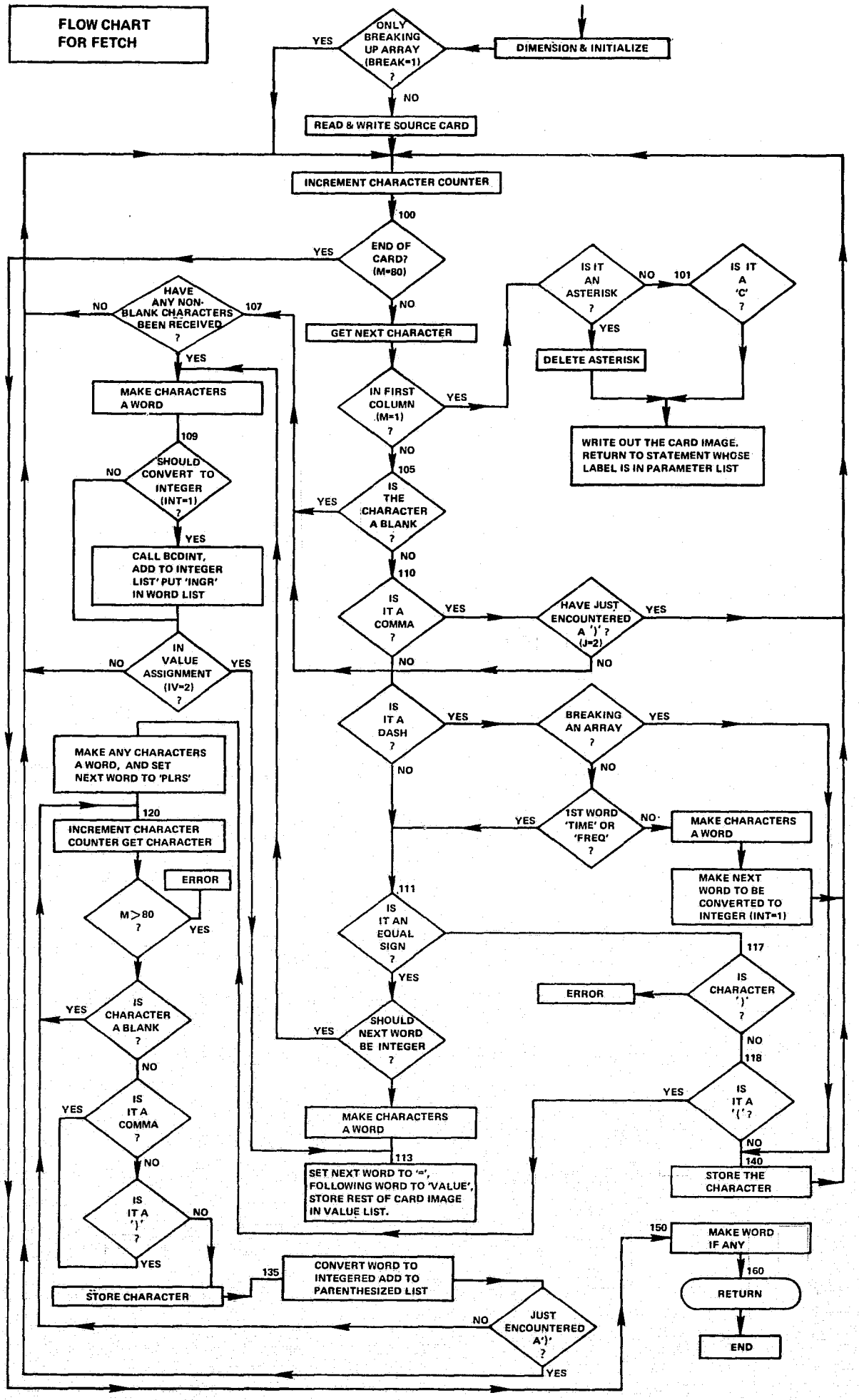


Figure III.3 Flow-Chart for Fetch

changed to integer and stored in a list. Every word is passed through the WORD array back to the calling program.

3.2 Syntactic Analysis

Syntactic analysis is the compiler subtask in which the major portion of the validity checking of source language statements occurs. It is also the function of this subtask to translate source language statements into equivalent FORTRAN statements which make up the source master control program.

Syntatic analysis is performed by FETCH, NTWORK, and the compiler main program. FETCH gives error messages for missing left and right parentheses, and any word that is supposed to be changed to integer that cannot be changed. Subroutine NTWORK checks the syntax of individual network elements. NTWORK also makes sure each element description card contains 'NODE' followed by node numbers, an element name, identification number, and a value. It also checks for valid terminal and port block syntax. Special checks are made to ensure that elements have the correct number of nodes specified. NTWORK checks to determine that special devices have been specified in the external model description part of the program. The main program checks the syntax of overall program structure.

The principal method used for the syntactic analysis of source program statements is the "sieve", or "keyword" method, in which the kind of source statement is first recognized by searching for certain keywords such as "NETWORK DESCRIPTION", "EXTERNAL MODEL", "AC", "DC", etc. After a keyword is recognized a subroutine (or equivalent) appropriate for the detailed syntactic analysis of that kind of source statement is activated. This subroutine outputs error messages (if any) in addition to the translated (object) form of the source statement.

The main program of the compiler forms the framework for the compilation system. The circuit design language compiler is written in FORTRAN, and translates a program written in the circuit design language into an equivalent

FORTRAN object program, the Master Control Program. Subroutines and computational routines are provided to perform the necessary circuit analysis.

The primary functions of the compiler are:

(1). to translate the input program written in circuit design language. The program is read in statement by statement and stored in the memory. In order to achieve greater memory availability, thus increasing the capability of the system, dynamic storage allocation is used;

(2). to check the syntax of the input program. Upon encountering any syntactical error, if the error is not severe, warning messages will be issued, and default conditions applied, and the compilation process continues and execution will occur. If the error is severe, warning messages will be given, and the compilation process will continue. However, the program will not be executed;

(3). to output statements into the master control program that direct the main flow of the computation. It arranges for the setting of control variables that give an indication of when, for example, the circuit analysis program has entered the network description, or has performed a certain analysis and is ready for output;

(4). to initialize master control program variables that control the activation of various circuit analysis subroutines;

(5). to set up parameters for calling various subroutines and computational routines.

The structure of the main program and its subsidiary subroutines is shown in Figure III. 4. It shows the hierarchical relationship between the main program and subroutines, the calling sequence of the program and its control directory.

The main program calls subroutines that perform special functions. It calls NTWORK in order to accept network descriptions and external model set-ups. Upon a request for execution, the main program calls EXEC, which checks that the upper and lower limits and the increments of time

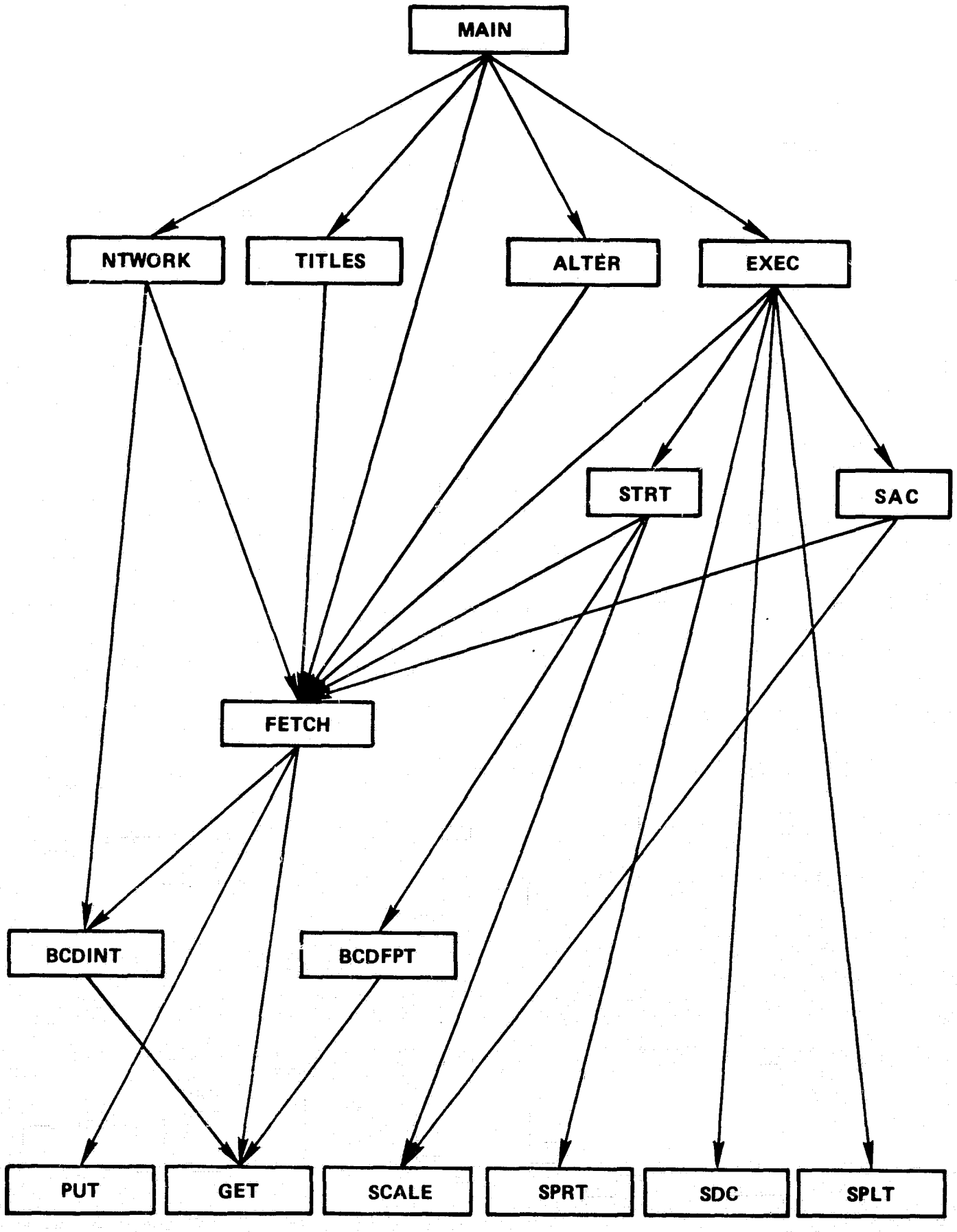


Figure III.4 Calling Hierarchy of Subroutines of Compiler

and frequency are defined. Subroutine EXEC calls SDC, SAC, or STRT depending on what type of analysis is requested. After the analysis is completed, SPRT and/or SPLT are called to print and/or plot the results of execution. Network elements can be modified for further analysis. Subroutine ALTER is called for this purpose.

FETCH is a subroutine which reads the input program and stores the information. It is called by the main program as well as by other subroutines whenever new statements must be read.

3.3 Circuit Topology and Data Storage

The storage of topology data is done by subroutine NTWORK, whose flow chart is given in Figure III.5. NTWORK reads a card by calling FETCH, then takes some action based upon the first word of the card. NTWORK operates in one of two modes. In its first mode, NTWORK processes network descriptions. In its second mode, NTWORK stores information for external model descriptions. In either mode, approximately the same actions occur.

The first test on the first word is to see if it is a port block or a terminal block. In either case, more cards are read until all of the external nodes have been input. For a terminal block, the input and output terminals are read and stored. If the first word on a card is 'NODE', a number of things can occur. First a test is made to see if the element is a resistor, inductor, capacitor, fixed voltage source, pulsed voltage source, sine-wave voltage source, pulsed current source, fixed current source, or impedance. If so, then a test is made to make sure that both nodes are present, and that an equal sign follows. The element name and identification number, both node numbers, and the name of the subnetwork in which the element appears are stored. If the element is one of the above except a pulsed or sine wave voltage source, its value (obtained by FETCH) is stored. If a pulsed or sine-wave voltage source, code words are stored for its value, and FETCH is called to decompose the value into individual words. These words are

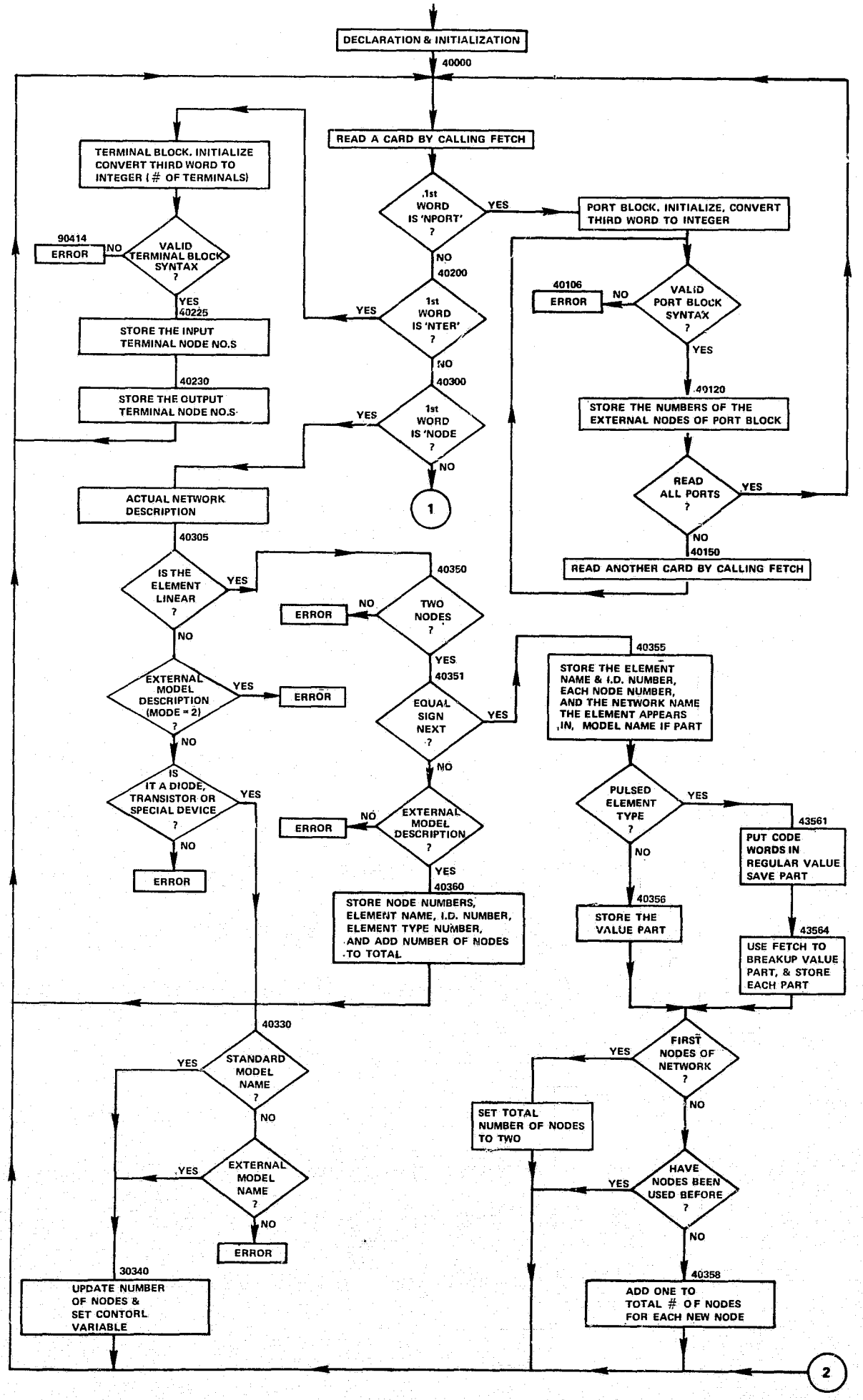


Figure III.5 Flow Chart of Subroutine NT Work

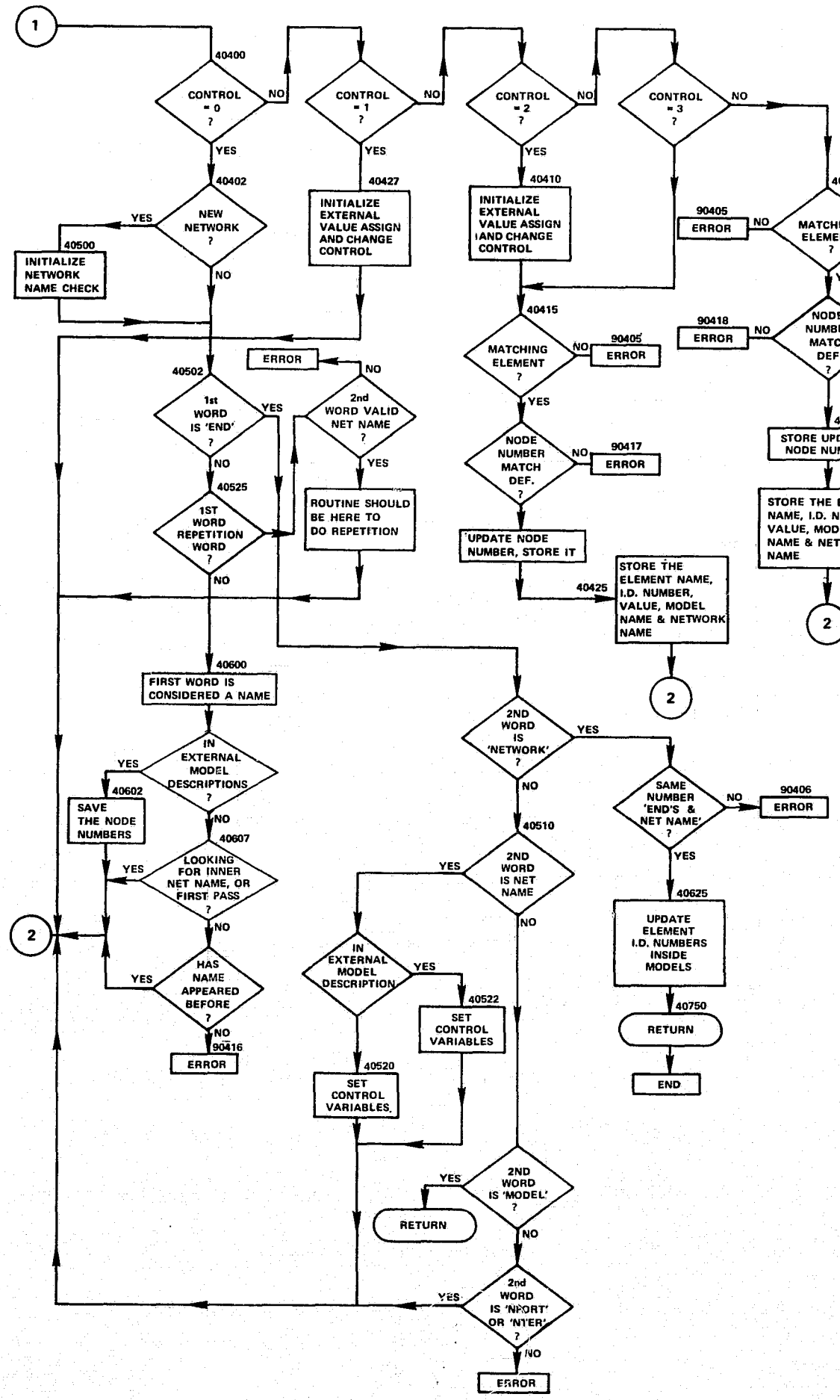


Figure III.5 Flow Chart of Subroutine NT Work (continued)

then saved for assignment. As element descriptions are input, the total number of nodes in the network is counted.

If the element is one of the aforementioned and no equal sign appears, then NTWORK should be in the external model description mode. In this case the element name, identification number and element type number are stored, and the total number of nodes in the external model is counted. FETCH is then reactivated to read the next card.

If the element is not one of the above, NTWORK checks to make sure it is in proper network description, for no allowance is made for nested nonlinear elements. If the element is not a diode, a transistor, or a special device, then an error has occurred. NTWORK then checks to make sure the element has a valid standard or external model name. If so, then the node numbers are saved and another card is read.

If the first word is not ' NODE' , then the next action depends upon the value of a control variable. If zero, then the card is checked to see if it is a new network name, a repeated name, or an ' END' statement. If an ' END' statement, then tests are made to determine what is ENDed. If the second word is ' NETWORK' , then this is the last statement of a network description. A test is made to make sure there are the corresponding number of subnetwork names and END statements. Then the elements in the stored list that are part of a model have their identification numbers updated (since they must be sequential), and NTWORK is exited. If the word after ' END' is a network name, then certain control variables can initiate a search for a new name, and the next card is read. If NTWORK is in the external model description mode, and the second word of an encountered END statement is a network name, then the end of a particular external model description is indicated. Control variables are set, and the next card is read. If the second word is ' NPORT' or ' NTER' , then this is the end of a port or terminal block. If the second word is none of these, then the programmer made an error.

If the first word of a statement is not ' END' , but is a repetition operator (i. e., ' REPEAT' , ' SERIES' , ' PARALL' , ' CASCAD'), NTWORK makes sure that the second word is a valid network name. This particular feature has not been implemented.

If the first word is not a repetition operator or ' END' , then it is assumed to be a network name. If this name occurs in an external model description, it is really the name of the external model, and the node numbers following this name are saved to be used when their elements must be updated.

4. Data

Interface Modules

Interface routines are included in the circuit design compiler to stored accept data from the compiler, transform it to fit the format of the CAPM' s, and store the new data in an appropriate data communication region. Two common features of the CAPM's are that they are all coded in FORTRAN, and all have a subroutine organization. Therefore the data can be stored and passed to these subroutines in three ways: through COMMON storage, through a parameter list, or both. The job of the interface routine is to set up these COMMON areas and argument lists before the master control program calls the execution routines of the CAPM. To transform the stored data from the compiler, it is useful to have utility routines that do concatenation, deconcatenation, conversion from character to integer, integer to character, and text editing. For example, the user program string ' R-1' is stored as character ' R' and integer 1, and the interface routine must convert integer 1 to character ' 1' , and concatenate it with character ' R' , producing ' R1' , the correct form for interface with the CIRCUS CAPM.

For example, consider CIRINT, the interface subroutine for the CIRCUS CAPM. CIRCUS is written in FORTRAN and uses COMMON storage to pass subroutine arguments. Thus the first section of CIRINT contains declarations and COMMON blocks exactly the same as those in CIRCUS.

The CIRCUS main program also contains these common blocks and declarations, and simply calls two subroutines, LINK2 and MAIN2. LINK2 initializes CIRCUS, reads input data, verifies its format, stores it, and executes some of the circuit analysis. MAIN2 does the rest of the analysis and writes out the results. To incorporate CIRCUS into the circuit design system, it was necessary to split LINK2 in half, with the first part being done by the compiler. The circuit analysis done by LINK2 could have been inserted at the end of CIRINT, but a new LINK2 was formed to perform these tasks, although CIRINT does the initialization of the program variables. The new LINK2 is called by CIRINT.

CIRCUS has four main storage areas besides the COMMON area. The COMMON block is dimensioned 16000 words. The first 216 words are for variables used internally in the program. The first storage area starts at location 217 and includes subsequently higher locations. This area contains the name and identification number of network elements, and control words (e.g., EXECUTE, PRINT). The second area starts at location 15000 and includes subsequently lower locations. This area contains pointers to element names and identification numbers, node numbers, element values, and repetition numbers (how many times elements are repeated). The third storage includes locations 15001 through 15500. This area contains element type codes in the order in which elements appear. The fourth storage area includes locations 15501 through 16000, and contains pointers that partition the information contained in the second storage area, in the order that elements occur.

The job of CIRINT is to set up these storage areas correctly so that the new LINK2 subroutine can be called, causing the CIRCUS CAPM to execute the correct analysis of the network. CIRCUS includes all elements except fixed current sources, but this was added by simulating a fixed current source, replacing it with a pulsed current source that has the same amplitude and a very long pulse duration. Another interface problem is that a resistor with identification number one would be stored in the system as character ' R'

in one word, and integer one in the next. In CIRCUS, this would be stored in the first storage area as character 'R1'. To perform the necessary conversion, a subroutine INTBC exists to convert a word from integer into character representation. With INTBC, and a combination of GET and PUT, a section of CIRINT accomplishes the correct transformation and storage. In the second area of storage, the pointers and node numbers presented no problem. However, the value of an element is stored as a character string in the system. Therefore it is necessary to use BCDFPT, a subroutine which changes a character string into an equivalent floating point number.

After CIRINT does the network topology data storage, it adds execution codes to the end of the first storage area. If printed output is requested, CIRINT adds the print commands to the first storage area. It then does the standard initialization of variables, and the initialization of the variables that depend on input from the system. CIRINT then calls the new LINK2, MAIN2, and then returns.

5. System Extendibility

One of the principal features of the Circuit Design System is its extendibility. The system was designed to provide a common user interface through a circuit design language to a number of circuit analysis packages. It was very quickly recognized that during the lifetime of the Circuit Design System, users would desire the addition of new or modified circuit analysis packages to the system. Accordingly, such extendibility was investigated and provided for in the design of the system.

There are two ways in which extendibility can be incorporated. The first method requires the user to provide his own data interface routine. The user must therefore completely understand the inner workings of the circuit analysis program module (CAPM) that he wishes to add to the circuit analysis system. This requires a user with some knowledge of programming, if in fact the user himself (rather than a professional programmer) will implement the data interface routine.

The other approach to system extendibility is essentially through text editing. It was found that the input formats of most circuit analysis packages were fairly similar, and the circuit design language and its associated compiler were designed to take advantage of this similarity. Accordingly, assuming that the circuit analysis package that the user wishes to add to the system has an input format not too dissimilar to the others in the system, text editing statements can be provided in the circuit design language to enable the user to directly convert input statements in circuit design language into the input format of his own package. This second method has the advantage of simplicity and can be readily employed by users with little or no programming experience. It has the disadvantages of being less general than the first method of extendibility, to somewhat bypasses the circuit design compiler.

There are subroutines supplied in the system that are useful in either method of extension. GET and PUT, described before, are useful for text editing. BCDFPT has the arguments 'ANS', 'BCD', 'N'. 'BCD' is an integer mode array containing characters to be converted into floating point mode. 'N' is the number of characters of BCD that are to be examined and converted into real numbers. BCDFPT will take any real number in character form, convert it to floating point, and put the results in the real variable 'ANS'. INTBC has integer arguments 'NUM', 'ICHAR', and 'L'. It takes the integer stored in the variable 'NUM', which should be less than 10,000, converts it into equivalent characters, and puts the character equivalent into 'ICHAR'. 'L' is used as an error check; if 'L' is zero after INTBC is called, no conversion took place. BCDINT is the inverse INTBC. It takes the double precision word in COMMON labeled INTBCD, converts that character representation of a number into an equivalent integer, and puts the result in BCDINT's only argument 'NUM'.

IV. SUMMARY AND CONCLUSIONS

The main objective of this investigation was to determine the feasibility of designing and implementing a circuit design system so that many circuit

analysis packages can be integrated through a common input language. Through a partial implementation of a proposed circuit design system, confidence was gained in the feasibility of this concept. Even though it is felt that a more complete implementation of the circuit design system would yield more complete answers, the preliminary implementation done thus far has worked quite well.

1. State of the Implementation

As discussed in some detail in Sections II and III, the circuit design language has been designed and specified, its compiler made operational, and a version of the entire circuit design system implemented and tested. Presently the only circuit analysis package included in the system is CIRCUS. The inclusion of additional packages presents no conceptual difficulty, and will require some additional programming effort. The implementation of generally specified nonlinear devices needs to be done. The methods of extendibility discussed in Section III. 5 have not been implemented.

2. Conclusions

Even though the project was not continued into its second year as anticipated, the preliminary work done during the first year has partially demonstrated the feasibility of a unified circuit design system. A more complete implementation would further demonstrate the concept of a unified circuit design system.

APPENDIX A

PROGRAM LISTING


```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                F01FEB69  5/30/70

                                     1          ENTRY     PUT
                                     2          ENTRY     GET
                                     3 GET      SAVE      (14,12),,*
000000 47F0 F008                 00008  4+GET    B         8(0,15) BRANCH AROUND ID
000004 03                               5+       DC        AL1(3)
000005 C7C5E3                           6+       DC        CL3'GET' IDENTIFIER
000008 90EC D00C                 0000C  7+       STM       14,12,12(13) SAVE REGISTERS
00000C 0520                             8        BALR      2,0
00000E                                  9        USING     *,2
00000E 5831 0000                 00000 10        L         3,0(1,0)
000012 5841 0004                 00004 11        L         4,4(1,0)
000016 5851 0008                 00008 12        L         5,8(1,0)
00001A 5844 0000                 00000 13        L         4,0(4,0)
00001E 5B40 2062                 00070 14        S         4,=F'1'
000022 1A34                             15        AR        3,4
000024 5860 2066                 00074 16        L         6,=CL4'    '
000028 5065 0000                 00000 17        ST        6,0(5,0)
00002C 4363 0000                 00000 18        IC        6,0(3,0)
000030 4265 0000                 00000 19        STC       6,0(5,0)
                                    20        RETURN    (14,12)
000034 98EC D00C                 0000C  21+       LM        14,12,12(13) RESTORE THE REGISTERS
000038 07FE                             22+      BR        14 RETURN
                                     23 PUT     SAVE      (14,12),,*
00003A 47F0 F008                 00008  24+PUT    B         8(0,15) BRANCH AROUND ID
00003E 03                               25+      DC        AL1(3)
00003F D7E4E3                           26+      DC        CL3'PUT' IDENTIFIER
000042 90EC D00C                 0000C  27+       STM       14,12,12(13) SAVE REGISTERS
000046 0520                             28       BALR      2,0
000048                                  29       USING     *,2
000048 5831 0000                 00000 30        L         3,0(1,0)
00004C 5841 0004                 00004 31        L         4,4(1,0)
000050 5851 0008                 00008 32        L         5,8(1,0)
000054 5844 0000                 00000 33        L         4,0(4,0)
000058 5B40 2028                 00070 34        S         4,=F'1'
00005C 1A34                             35        AR        3,4
00005E 4365 0000                 00000 36        IC        6,0(5,0)
000062 4263 0000                 00000 37        STC       6,0(3,0)
                                    38        RETURN    (14,12)
000066 98EC D00C                 0000C  39+       LM        14,12,12(13) RESTORE THE REGISTERS
00006A 07FE                             40+      BR        14 RETURN
                                    41        END
000070 00000001                         42               =F'1'
000074 40404040                         43               =CL4'    '
```

```
            C     PROGRAM----MAIN
            C             RICHARD CHAN    AUGUST 1969
            C
            C     THIS IS THE MAIN PROGRAM FOR CIRCUIT DESIGN PROGRAMMING LANGUAGE
            C     IT IS A PROGRAMMING SYSTEM WHICH TRANSLATES THE PROGRAM WRITTEN
            C     IN CIRCUIT DESIGN LANGUAGE INTO FORTRAN PROGRAM AND THEN USING
            C     FORTRAN COMPILER TO EXECUTE THE PROGRAM.
            C
0001              DOUBLE PRECISION WORD(30), DUMMY, PRLS,
                 1     EXTERL, END, MODEL, NETWOK, DESCRI, FREQUN, TIME, PRINT,
                 2  EXECUT, PLOT, MODIFY, DELETE, CHANGE, ADD, PROGRM, DC, AC,
                 4  TRANST, DPRO, BLANK
0002              DOUBLE PRECISION DLIST(2,100), PULVAL,          EXTRSV(20)
0003              DOUBLE PRECISION NTSV(7,7), CURRNT,VOLTGE
0004              DIMENSION NODESV(7), NCNTEL(7), ELIST(4, 100)
0005              DIMENSION MAXLST(10), INTEGR (10), VALUE(20), PRLIST(10, 15)
                 1  ,    LABEL( 9) ,  NOEXEL(20),IDSAVE(12)
0006              DIMENSION ELVALU(50,10),INPUT(5,10,2),OUTPUT(5,10,2),TRINPT(5,15),
                 1TROUPT(5,15), XTRLST(5, 30, 20), PULVAL(3,20,7)
0007              DIMENSION FREQ(12), IPCURR(2,100), IPVOLT(2,100)
0008              DIMENSION ZDUMN(1600)
0009              COMMON ZDUMN
0010              COMMON/CAL/ ELIST,ELVALU,XTRLST,INPUT,OUTPUT,TRINPT,TROUPT,DLIST
0011              COMMON /INTBCD/ DUMMY
0012              COMMON/VBREAK/ ICUMM(21)
0013              COMMON/PRINTE/ IPCURR, IPVOLT, NELMNT
0014              COMMON/TIMINT/ FREQ
0015              INTEGER  PRLIST, VALUE, STOPER, STAR, C ,  TYPE, XTRLST
0016              INTEGER ELVALU,OUTPUT,TRINPT,TROUPT, ELIST
0017              DATA  STAR/'*'/,        C/'C'/ ,        EXTERL/'EXTERNAL'/,
                 1      END/'END     '/,  MODEL/'MODEL   '/,  NETWOK/'NETWORK '/,
                 2      DESCRI/'DESCRIPT'/,  FREQUN/'FREQUNCY'/,  TIME/'TIME    '/,
                 3      PRINT/'PRINT   '/,  EXECUT/'EXECUTE '/,  PLOT/'PLOT    '/,
                 4   . MODIFY/'MODIFY  '/,  DELETE/'DELETE  '/,  CHANGE/'CHANGE  '/
                 5   , ADD/'ADD     '/,  PROGRM/'PROGRAM '/,  DC/'DC      '/,
                 6      AC/'AC      '/,  TRANST/'TRANSIEN'/,  BLANK/'       '/,
                7CURRNT,VOLTGE/'CURRENT','VOLTAGE'/
0018              DATA N9999/'9999'/
0019              DO 11 I=1, 9
0020        11    LABEL(I)=I*10000
0021              DO 13 I = 1,12
0022              FREQ(I) = 0
0023        13    IDSAVE(I) = 0
0024              STOPER = 0
0025              NUMEXT=0
0026              IPROM=1
0027        11111 ICARD=0
0028              WRITE (6, 8)  IPROM
0029        8     FORMAT (' C'/' C'/' C     THIS IS THE OUTPUT PROGRAM FOR PROBLEM N
                 10. ', I5)
            C     INTIATE THE NUMBER OF CARD READS FOR EACH PROBLEM
0030              ISECTN = 1
            C     READ IN ONE CARD AT EACH TIME BY 'FETCH'
0031        10000 CALL FETCH (WORD, PRLIST, MAXLST, NWORD, VALUE, NPRLST ,
```

```
                1  STOPER, INTEGR , ICARD, &10000,0)
0032            ICARD=ICARD+1
0033            IF (ICARD  .GT.  1 )  GO TO 10010
          C     FIRST CARD IN EACH PROBLEM IS ASSUMED TO BE PROBLEM IDENTIFIER
          C     IT IS READ AND OUTPUTED AS TITLE LINE FOR EACH CIRCUOT PROBLEM
0034            IF (WORD(1)  .EQ. END  .AND.  WORD(2)  .EQ.  PROGRM )  GO TO 99999
0035            DPRO=WORD(1)
0036            LABEL1=LABEL(1)
0037            CALL TITLES(IPRCM, DPRO, LABEL1, ICARD)
0038            LABEL(1)=LABEL1
0039            GO TO 10000
0040      10010 IF (WORD(1)  .EQ. STAR  .OR.  WORD(1)  .EQ. C )  GO TO 10000
          C     STAR CARD IS A VALID FORTRAN STATEMENT, AND C CARD IS A COMMETCARD
          C     THEY WILL BE OUTPUTED DIRECTLY WITHOUT TRANSLATION
          C     IT IS HANDLED BY FETCH SUBROUTINE
0041            IF (ISECTN  .GT.  1 )  GC TO 30000
          C     THE FIRST SECTION IS FOR EXTERNAL MODEL
0042            IF(WORD(1)  .EQ.  EXTERL  .AND.  WORD(2)  .EQ.  MODEL) GO TO 10020
0043            IF (WORD(1)  .EQ.  NETWOK )  GO TO  20001
0044            IF (WORD(2)  .EQ.  DESCRI )  GO TO  20001
0045            STOPER = 1
0046            WRITE (6, 80002)
0047      80002 FORMAT (1X, 'C     **MISSING EXTERNAL MODEL CARD**  THIS CARD READ
                1S')
0048            WRITE (6, 80003) (WORD(I), I=1, NWORD)
0049      80003 FORMAT (6X, 10A8)
0050      20000 ISECTN=ISECTN+1
0051            MODE=2
          C     SUBROUTINE NET IS USED FOR ACCEPTING EITHER THE EXTERNAL MODEL OR
          C     NETWORK SETUP.  IT IS CONTROLLED BY 'MODE' AND 'STOPER'
0052      40010 CALL       NTWORK(MODE, EXTRSV, NOEXEL, NUMEXT, ICARD, STOPER,
                1    NODESV, NTSV, NONTEL, NELMNT,PULVAL,IDSAVE)
0053            ISECTN=ISECTN+1
0054            DO 20 K=1, 7
0055            KK=8-K
0056            IF (NTSV(KK, 1)  .EQ.  0)  GO TO 20
0057            DO 15 J=1, 6
0058            JJ=8-J
0059            IF (NTSV(KK, JJ)  .EQ.  0)  GO TO 15
0060            DO 10  L=1, 7
0061            IF (NTSV(L, 1)  .EQ.  NTSV(KK, JJ))  GO TO 12
0062      10    CONTINUE
0063      12    NODESV(KK)=NODESV(KK)+NODESV(L)
0064      15    CONTINUE
0065      20    CONTINUE
          C     STOPER=1  IS CONSIDERED AS EITHER MISSING EXTERNAL MODEL CARD OR
          C     NETWORK DESCRIPTION CARD
          C     WARNING IS PROVIDED ANC THE PROGRAM WILL BE CONTINUED
          C     WORD READ IN THIS CARD IS THEN PASSED TO SUBROUTINE NET
0066            GO TO 10000
0067      10020 WRITE (6, 80004) (WORD(I), I=1, 2), ICARD, ISECTN
          C     THIS IS TO WRITE EXTERNAL MODEL CARD
0068      80004 FORMAT (1X, 'C     ', 2A8, 10X, 'CALL SUBROUTINE NTWORK AT THIS PO
                1INT',  35X, 2I6)
```

```
0069            GO TO 20000
0070      20001 ISECTN=ISECTN+2
0071            GO TO 30020
                C     FINISH EXTERNAL MODEL SECTION, NET WILL READ THE END MODEL CARD
                C     AND INCREMENT ISECTN BY 1
                C     NEXT SECTION IS FOR NETWORK DESCRIPTION
                C     ISECTION RETURNED FROM NET AFTER THE END MODEL CARD BEEN READ =3
0072      30000 IF (ISECTN .GT. 3 ) GOTO 50000
0073            IF (WORD(1) .EQ. NETWOK ) GO TO 30020
0074            IF ( WORD(2) .EQ. DESCRI ) GO TO 30020
0075            STOPER = 1
0076            WRITE (6, 80005)
0077      80005 FORMAT (1X, 'C     **MISSING NETWORK DESCRIPTION CARD**  THIS CARD
                1 READS')
0078            WRITE (6, 80003) ( WORD(I), I=1, NWORD)
0079      40000 ISECTN = ISECTN+1
0080            MODE=1
0081            GO TO 40010
0082      30020 WRITE (6, 80004) ( WORD(I), I=1, 2), ICARD, ISECTN
                C      THIS TO WRITE OUT NETWORK DESCRIPTION CARD
0083            GO TO 40000
                C     FINISH NETWORK SETUP SECTION, NET WILL READ THE END NETWORK CARD
                C     AND INCREMENT ISECTN BY 1
                C     NEXT SECTION IS FOR COMMAND SECTION WHICH SPECIFIES THE TYPE OF
                C     ANALYSIS, WHAT KIND OF OUTPUT AND ANY MODIFICATION TO BE MADE
                C
0084      50000 IF (WORD(1) .EQ. EXECUT ) GO TO 60001
0085      50010 IF (WORD(1) .EQ. MODIFY ) GO TO 60050
0086            IF (WORD(1) .EQ. DELETE ) GO TO 60056
0087            IF (WORD(1) .EQ. CHANGE ) GO TO 60060
0088            IF (WORD(1) .EQ. ADD ) GO TO 60066
0089            IF (WORD(1) .NE. END ) GO TO 61000
0090            IF (WORD(2) .EQ. DPRO ) GO TO 62000
0091            IF (WORD(2) .EQ. PROGRM ) GO TO 99999
0092      89000 WRITE (6, 89999 ) ( WORD (I), I=1, 10)
0093      89999 FORMAT (1X, 'C     ERROR ENDING, ASSUME TERMINATED OF PROGRAM'/1X,
                1 'C     ', 10A8)
0094            GO TO 99999
0095      60001 TYPE=0
0096            IF (WORD(2) .EQ. DC ) TYPE=3
0097            IF (WORD(2) .EQ. AC ) TYPE=1
0098            IF (WORD(2) .EQ. TRANST ) TYPE=2
0099            IF ( TYPE .GT. 0 ) GO TO 60022
0100            STOPER = 1
                C     ANALYSIS SPECIFICATION CARD ERROR, THE ANALYSIS WILL NOT BE
                C     PROCEEDED
0101            WRITE (6, 80006)
0102      80006 FORMAT (1X, 'C     **ERROR INPUT CARD**  THIS CARD READS')
0103            WRITE (6, 80007) (WORD(I), I=1, NWORD)
0104      80007 FORMAT (1X, 'C     ', 10A8)
0105            WRITE (6, 80010)
0106      80010 FORMAT (' C     ** THIS SECTION WILL BE IGNORED DUE TO THE ERROR
                1IN PREVIOUS INPUT **')
0107      60020 CALL FETCH (WORD, PRLIST, MAXLST, NWORD, VALUE, NPRLST ,
```

```
                1  STOPER, INTEGR , ICARD, &60020,0)
0108            ICARD=ICARD+1
0109            IF (WORD(1)  .EQ.  FREQUN )  GO TO  60030
0110            IF (WORD(1)  .EQ.  TIME   )  GO TO  60030
0111            IF (WORD(1)  .EQ.  PRINT  )  GO TO  60030
0112            IF (WORD(1)  .EQ.  PLOT   )  GO TO  60030
0113            IF (WORD(1)  .EQ.  EXECUT )  GO TO  60001
0114            WRITE (6, 80020)
0115      80020 FORMAT (1X, 'C     ** ERROR FOUND IN THIS CARD ** CARD READS ')
0116            WRITE (6, 80007) ( WORD(I), I=1, NWORD)
0117            GO TO 50010
0118      60022 CALL EXEC (TYPE, WORD, NWORD, LABEL, ICARD, NTSV, NODESV, NONTEL,
                1  &50000)
0119            GOTO 10000
0120      60030 WRITE (6, 80007) (WORD(I), I=1, NWORD)
0121            GO TO 60020
0122      60050 NALTER=1
0123      60051 CALL ALTER (NALTER, WORD, NWORD, PRLIST, ICARD)
0124            GO TO 10000
0125      60056 NALTER=2
0126            GO TO 60051
0127      60060 NALTER = 3
0128            GO TO 60051
0129      60066 NALTER =4
0130            GO TO 60051
0131      61000 STOPER =2
          C     UNIDENTIFY CARD
0132            WRITE(6, 80006)
0133            WRITE (6, 80007) (WORD(I), I=1, NWORD)
0134            IF (WORD(2)  .EQ.  DC  .OR.  WORD(2)  .EQ.  AC  .OR.  WORD(2) .EQ.
                1  TRANST )  GO TO  60020
0135            IF (WORD(1)  .EQ.  PRINT  .OR.  WORD(1)  .EQ.  PLOT  .OR.  WORD(1)
                1  .EQ.  FREQUN  .OR.  WORD(1)  .EQ.  TIME )  GO TO 60030
0136            GO TO 89000
0137      62000 IPROM=IPROM+1
0138            STOPER =0
0139            GO TO 11111
0140      99999 CONTINUE
0141            NONODE = NODESV(1)
0142      4     READ(5,5)  (VALUE(I),I=1,20)
0143      5     FORMAT(20A4)
0144            IF( VALUE(17) .EQ. N9999) GO TO 6
0145            WRITE(11,5)(VALUE(I),I=1,20)
0146            GO TO 4
0147      6     WRITE(6,1) NELMNT, NONODE
0148            WRITE(10,1)NELMNT, NONODE
0149      1     FORMAT(I4,I4)
0150            WRITE(10,7)((ELIST(I,J),I = 1,4),J = 1,NELMNT)
0151      7     FORMAT(2I4,A4,I4)
0152            WRITE(10,14)((ELVALU(I,L),L=1,10),I=1,NELMNT)
0153      14    FORMAT(2(10A4) )
0154            WRITE(10,16)(ICSAVE(I),I=1,12)
0155      16    FORMAT( 12I5)
0156            WRITE(10,21)(FREQ(I),I=1,12)
0157      21    FORMAT( 6(E8.3,2X),20X )
0158            WRITE(10,26)( ( IPCURR(I,J),I=1,2),J=1,100)
0159            WRITE(10,26)( ( IPVOLT(I,J),I=1,2),J=1,100)
0160      26    FORMAT(  10(A4,I4) )
0161            WRITE(10,28) ((( PULVAL(I,J,K),K= 1,7),J=1,20),I=1,3)
0162      28    FORMAT( 7(A8,2X) )
0163            STOP
0164            END
```


```
0001                SUBROUTINE FETCH(WORD, PRLIST, MAXLST, NWORD,VALUE,NPRLST,
                   1  STOPER, INTEGR , ICARD, *,BREAK)
               C    READS A CARD IN A FREE FEILD MANNER
0002                INTEGER   BLANK, COMMA, EQUAL, LPAREN, RPAREN, ASTRIX, CEE, DECPT
0003                DOUBLE PRECISION DUMMY, WORD(30), INGR, TIME, FREQ
                   1, CLEAR,PRLS,VALU,DEQUAL,EXPO
0004                INTEGER BUFF1, BUFF2 ,PRLIST,TEST,DASH
0005                COMMON/VBREAK/BUFF1, IBREAK
0006                COMMON  /INTBCD/DUMMY
0007                INTEGER STOPER,VALUE,PRLIST, BREAK
0008                DIMENSION MAXLST(10),BUFF1(20),PRLIST(10,15),INTEGR(10),VALUE(20)
0009                DATA      BLANK, COMMA, EQUAL, LPAREN, RPAREN, ASTRIX, CEE, DECPT
                   1/      ' ', ',', '=', '(', ')', '*', 'C',',','/,DASH/'-'/
0010                DATA VALU,DEQUAL,EXPO/          'VALU    ','=       ','**      '/
0011                EQUIVALENCE (BUFF2,DUMMY)
0012                DATA CLEAR/'        '/,INGR/'INGR    '/,PRLS,NZERO,NINE/'PRLS    ',
                   1'0   ','9   '/ , TIME, FREQ/'TIME    ', 'FREQUNCY'/
0013                DO 90 I = 1,30
0014          90    WORD(I) = CLEAR
0015                DO 92 I = 1,10
0016                DO 91 J = 1,15
0017          91    PRLIST(I,J) = 0
0018                INTEGR(I) = 0
0019          92    MAXLST(I) = 0
0020                DO 93 I = 1,20
0021          93    VALUE(I) = 0
0022                IBREAK = 0
0023                III = 0
0024                INT = 0
0025                IP = 0
0026                IV = 1
0027                KKK = 0
0028                L   =0
0029                LI = 0
0030                M   =0
0031                N   =0
0032                DUMMY = CLEAR
0033                IF ( BREAK .EQ. 1) GO TO 99
0034                READ(5,1001)  (BUFF1(I), I = 1,20)
0035          1001  FORMAT(20A4)
0036                WRITE(6,1002)ICARD,(BUFF1(I),I=1,20)
0037          1002  FORMAT(30X,'CARD #',I3,' = ',20A4)
0038          99    II = 1
0039          100   IF ( M.EQ. 80)  GO TO 150
0040                M=M+1
0041                CALL GET(BUFF1,M,TEST)
               C    GET  PLACES THE M-TH CHARACTER OF BUFF1 LEFT-ADJUSTED INTO TEST
0042                IF (M .NE. 1)      GO TO 105
               C    CHECKING FOR AN ASTRISCK OR A 'C' IN COLUMN 1
0043                IF (TEST .NE. ASTRIX)      GO TO 101
0044                WORD(1) = ASTRIX
0045                CALL PUT(BUFF1,1,BLANK)
0046                WRITE(6,1007) (BUFF1(I), I=1,20)
0047          1007  FORMAT(1X,20A4)
```

```
0048            RETURN 1
0049       101  IF(TEST .NE. CEE)      GO TO 105
0050            WORD(1) = CEE
0051            WRITE(6,1007) (BUFF1(I), I=1,20)
0052            RETURN 1
0053       105  IF (TEST .NE. BLANK)      GO TO 110
0054       107  IF( N .EQ. 0) GO TO 100
0055            L = L + 1
0056            WORD(L) = DUMMY
0057            IF( INT .NE. 1) GO TO 109
0058       108  WORD(L) = INGR
0059            CALL BCDINT(NUM)
0060            LI = LI + 1
0061            INTEGR(LI) = NUM
0062            INT = 0
0063       109  DUMMY = CLEAR
0064            N = 0
0065            GO TO (100,112),IV
0066       110  IF(TEST .EQ. COMMA)      GO TO (107,99),II
0067            IF(BREAK .EQ. 1) GO TO 140
0068            IF ( TEST .NE. DASH) GO TO 111
0069            IF (WORD(1)  .EQ.  TIME  .OR.  WORD(1)  .EQ.  FREQ )  GO TO 111
0070            L = L + 1
0071            INT = 1
0072            WORD(L) = DUMMY
0073            DUMMY = CLEAR
0074            GO TO 100
0075       111  IF(TEST .NE. EQUAL)      GO TO 117
                C
                C          **      VALUE ASSIGNMENT       **
                C      HAVE ENCOUNTERED'='; IF N > 0 STORE BUFF2 IN NEXT WORD,'=' IN
                C       FOLLOWING WORD. IF N = 0 STORE '=' IN NEXT WORD.
                C      AFTER ''='', NEXT WORDS ARE STORED IN VALUE, NEXT WORD = 'VALUE'.
0076            IF( N .EQ. 0) GO TO 112
0077            IF( INT .NE. 1) GO TO 1111
0078            L = L + 1
0079            IV = 2
0080            GO TO 108
0081       1111 L=L+1
0082            WORD(L) = DUMMY
0083            DUMMY = CLEAR
0084            N = 0
0085       112  L = L + 1
0086            WORD(L) = DEQUAL
0087            L = L + 1
0088            WORD(L) = VALU
0089            ITEST = MOD(M,4)
0090            IF( ITEST .EQ. 0) GO TO 115
0091            I = M - ITEST
0092            DO 113 K = I,M
0093       113  CALL PUT(BUFF1,K,BLANK)
0094       115  IBEGIN = (M - ITEST)/4 + 1
0095            J = 0
0096            DO 116  I = IBEGIN,20
```

```
0097                J = J + 1
0098          116   VALUE(J) = BUFF1(I)
0099                GO TO 150
0100          117   IF(TEST .NE. RPAREN) GO TO 118
0101                WRITE(6,1040) ICARD
0102          1040  FORMAT ('C     ', 5X, '***** MISSING A LEFT PARENTHISIS ON CARD',
                   1  I5, ' *****')
0103                STOPER = 1
0104          118   IF(TEST .NE. LPAREN) GO TO 140
                    C
                    C       **     PARENTHSIZED LIST ASSIGNMENT     **
                    C     HAVE RUN INTO A PARENTHSIZED PARAMETER LIST. WILL SUBSTITUTE
                    C     'PRLS' FOR WORD, STORE THE PARAMETERS INTO  PRLIST(I,J).
0105                J = 0
0106                II = 1
0107                IF(N .EQ. 0) GO TO 119
0108                L = L + 1
0109                WORD(L) = DUMMY
0110                DUMMY = CLEAR
0111                N = 0
0112          119   IP = IP + 1
0113                L = L + 1
0114                WORD(L) = PRLS
0115          120   M = M + 1
0116                IF ( M .NE. 81 ) GO TO 121
0117                WRITE(6,1035) ICARD
0118          1035  FORMAT ('C     ', 5X, '***** MISSING A RIGHT PARENTHISIS ON CARD',
                   1  I5, ' *****')
0119                STOPER = 1
0120                GO TO 150
0121          121   CALL GET(BUFF1,M,TEST)
0122                IF (TEST .EQ. BLANK)      GO TO 120
0123                IF(TEST .EQ. COMMA)       GO TO 135
0124                IF(TEST .EQ. RPAREN) GO TO 130
0125                N = N + 1
0126                CALL PUT(BUFF2,N,TEST)
0127                GO TO 120
0128          130   II = 2
0129          135   J = J + 1
0130                CALL BCDINT(NUM)
0131                PRLIST(IP,J) = NUM
0132                DUMMY = CLEAR
0133                N = 0
0134                MAXLST(IP) = J
0135                GO TO (120,100),II
                    C
                    C       **   CHARACTER INSERTION   **
                    C     CHARACTER IS NOT A DELIMETER
0136          140   N = N + 1
0137                II = 1
0138                CALL PUT(BUFF2,N,TEST)
0139                GO TO 100
0140          150   IF(N .EQ. 0) GO TO 160
0141                L = L + 1
0142                WORD(L) = DUMMY
0143          160   NWORD = L
0144                NPRLST = IP
0145                RETURN
0146                END
```

```
FORTRAN IV G LEVEL 1, MOD 4          EXEC          DATE = 70150          15/07/10          PAGE 0001

0001            SUBROUTINE EXEC ( TYPE, WORD, NWORD, LABEL, ICARD, NTSV,
               1     NCDESV, NONTEL, *)
          C     THIS IS A SUBROUTINE USED TO EXECUTE THE ANALYSIS CALLED FROM THE
          C     MAIN PROGRAM
          C
0002            DIMENSION MAXLST(10), INTEGR (10), VALUE(20), PRLIST(10, 15)
0003            COMMON/INTBCD/DUMMY
0004            INTEGER  STOPER, TYPE, VALUE, PRLIST
0005            DOUBLE PRECISION PRINT, PLOT
0006            DOUBLE PRECISION  NTNAME, ANALYS
0007            DIMENSION LABEL( 9), NCDESV(7), NONTEL(7)
0008            DOUBLE PRECISION  NTSV(7,7)
0009            DOUBLE PRECISION WORD(30), DUMMY, PRLS
0010            DATA PRINT, PLOT/'PRINT   ', 'PLOT    '/
0011            ANALYS=WORD(2)
0012            NTNAME=WORD(3)
0013            IOUT=0
0014            GO TO ( 60011, 60021, 60031 ), TYPE
0015       60011 CALL SAC    (  TYPE, WORD, NWORD, LABEL, ICARD)
0016            GO TO 100
0017       60021 CALL STRT   (  TYPE, WORD, NWORD, LABEL, ICARD)
0018            GO TO 100
0019       60031 CALL SDC    (  TYPE, WORD, NWORD)
0020       100  CALL FETCH (WORD, PRLIST, MAXLST, NWORD, VALUE, NPRLST,
               1  STOPER, INTEGR, ICARD, &100,0)
0021            ICARD=ICARD+1
0022            IF ( WORD(1)  .EQ.  PRINT )  IOUT=1
0023            IF ( WORD(1)  .EQ.  PLOT  )  IOUT=2
0024            IF (IOUT  .GT.  0 )  GO TO 101
0025            WRITE (6, 840) ICARD
0026       840  FORMAT (1X, 'C     **ERROR OUTPUT SPECIFICATION CARD IN NO. ', I5,
               1  '**')
0027            GO TO 999
0028       101  GO TO (102, 103, 104), IOUT
0029       102  CALL SPRT (ANALYS, WORD, NWORD, NTNAME, LABEL, ICARD, NTSV
               1 ,NODESV, NONTEL)
0030            GO TO 105
0031       103  CALL SPLT ( TYPE, WORD, NWORD )
0032       105  IOUT=3
0033            GO TO 100
0034       104  RETURN1
0035       999  RETURN
0036            END
```

```
FORTRAN IV G LEVEL 1, MOD 4          SPLT          DATE = 70150          15/07/10          PAGE 0001

0001            SUBROUTINE SPLT (TYPE, WORD, NWORD)
          C     SUBROUTINE
          C
          C
0002            DOUBLE PRECISION WORD(30), BLANK
0003            INTEGER  STOPER, TYPE
0004            DATA BLANK/'        '/
0005            DO 50  I=2, NWORD
0006            IF (WORD(I)  .EQ. BLANK )  GO TO 50
0007            WRITE (6, 88)  WORD(I)
0008       50   CONTINUE
0009       88   FORMAT (6X, 'PLOT ', 2X, A8)
0010            RETURN
0011            END
```

```
0001            SUBROUTINE TITLES(IPROM, DPRO, LABEL1, ICARD)
0002            DOUBLE PRECISION WORD(30), DUMMY, PRLS, END, DECLAR
0003            DOUBLE PRECISION DPRO
0004            DIMENSION MAXLST(10), INTEGR (10), VALUE(20), PRLIST(10, 15)
0005            INTEGER  PRLIST, VALUE, STOPER
0006            COMMON/INTBCD/DUMMY
0007            DATA END/'END     '/, DECLAR/'DECLARE '/
0008            WRITE (6, 80000)  DPRO
0009       80000 FORMAT (' C      PROGRAM IDENTIFIER IS ', A8)
0010            K=1
0011       100  CALL FETCH (WORD, PRLIST, MAXLST, NWORD, VALUE, NPRLST ,
                1  STOPER, INTEGR, . ICARD,   &100,0)
0012            ICARD=ICARD+1
0013            IF (WORD(1)  .EQ.  DECLAR )  GO TO 200
0014            IF (WORD(1)  .EQ.  END  .AND.  WORD(2) . .EQ. DECLAR )  GO TO 300
0015            IF ( K  .LE.  1 )  GO TO 300
0016            GO TO 100
0017       200  K=2
0018            WRITE (6, 80005)
0019       80005 FORMAT (' C      ** DECLARTION SECTION **')
0020            GO TO 100
0021       300  WRITE (6, 80002)
0022       80002 FORMAT (7X, 'COMPLEX IMPDEN, ADMITN'/
                1  7X, 'REAL INDUCT'/
                2  7X, 'REAL MU'/
                3  7X, 'DIMENSION FREQ(1000), TIME(1000), RESIST(1000),CAPACT( 1000
                4),'/17X, 'INDUCT(1000), CURRET(1000), VOLTAG(1000), IMPDEN(1000),'
                5 / 17X, 'ADMITN(1000)' )
0023             WRITE (6, 80001) LABEL1, LABEL1
0024       80001 FORMAT (' C      ** INITIATION SECTION **'/
                A        7X, 'DO ', I5, ' I=1, 1000'/7X, 'CAPACT(I)=1'/ 7X,
                B  'FREQ(I)=0'/
                1 7X, 'TIME(I)=0'/ 7X, 'RESIST(I)=1'/ 7X, 'CAPACT(I)=1'/
                2  7X, 'INDUCT(I)=1'/ 7X, 'CURRET(I)=0'/ 7X,'VOLTAG(I)=0'/
                3 7X, 'IMPDEN(I)=(1, 0)'/ 1X, I5, 1X, 'ADMITN(I)=(1, 0)' )
0025             WRITE (6, 80004)
0026       80004 FORMAT (' C      ** START ACCEPTING THE NETWORK PROGRAM INPUT CARD
                1SETUP ** ')
0027             LABEL1=LABEL1+1
0028             RETURN
0029             END
```

```
FORTRAN IV G LEVEL 1, MOD 4          BCDINT          DATE = 70150          15/07/10          PAGE 0001

0001                SUBROUTINE BCDINT(NUM)
0002                INTEGER BCD,TEST,DIGIT,BLANK
0003                DOUBLE PRECISION RBCD
0004                COMMON  /INTBCD/RBCD
0005                EQUIVALENCE (BCD,RBCD)
0006                DIMENSION DIGIT(10)
0007                DATA DIGIT/'0','1','2','3','4','5','6','7','8','9'/,BLANK/'    '/
0008                NUM = 0
0009                DO 10 M = 1,8
0010                CALL GET(BCD,M,TEST)
0011                IF ( TEST .EQ. BLANK ) GO TO 10
0012                DO 20 K = 1,10
0013                IF ( TEST .EQ. DIGIT(K) ) GO TO 30
0014          20    CONTINUE
0015                WRITE(6,25) M, TEST, RBCD
0016          25    FORMAT(' POSITION # ',I2,' = ',A4,'IS NOT A DIGIT IN', A8)
0017                RETURN
0018          30    NUM = 10*NUM + K - 1
0019          10    CONTINUE
0020                RETURN
0021                END

FORTRAN IV G LEVEL 1, MOD 4          ALTER           DATE = 70150          15/07/10          PAGE 0001

0001                SUBROUTINE ALTER(NALTER, WORD, NWORD, PRLIST, ICARD)
              C     SUBROUTINE
0002                INTEGER  STOPER, TYPE, VALUE, PRLIST
0003                DOUBLE PRECISION WORD(30), PRLS, DUMMY, ALTE
0004                DIMENSION MAXLST(10), INTEGR (10), VALUE(20), PRLIST(10, 15)
0005                COMMON/INTBCD/DUMMY
0006                MFCARD=0
0007                ALTE =WORD(1)
0008          11    CALL FETCH (WORD, PRLIST, MAXLST, NWORD, VALUE, NPRLST,
                   1  STOPER, INTEGR , ICARD, &11,0)
0009                ICARD=ICARD+1
0010                IF (WORD(1)  .EQ.  END  .AND.  WORD(2)  .EQ.  ALTE  )  GO TO 1
0011                MFCARD=MFCARD+1
0012                WRITE (6, 80)  ALTE , MFCARD, (WORD(I), I=1, NWORD)
0013          80    FORMAT('0C     ', A8, 2X, I4, 10(1X, A8))
0014                GO TO 11
0015          1     WRITE (6, 81) ALTE
0016          81    FORMAT('0C    END OF ', A8)
0017                RETURN
0018                END

FORTRAN IV G LEVEL 1, MOD 4          SDC             DATE = 70150          15/07/10          PAGE 0001

0001                SUBROUTINE SDC(TYPE, WCRD, NWORD)
              C     SUBROUTINE
              C
              C     A COMPUTATIONAL ROUTINE WILL BE INTENDED TO BE ADDED INTO THE
              C     LANGUAGE, IT WILL BE APEARED IN WORD(4) IN THE ANALYSIS CARD.
              C
0002                DOUBLE PRECISION WORD(30), PWORD(30)
0003                INTEGER  STOPER, TYPE
0004                WRITE (6, 840)  WORD(3), WORD(4)
0005          840   FORMAT('0C      THIS IS A DUMMY SUBROUTINE, IT WILL BE SUBSTITUTED
                   1BY A DC ANALYSIS ROUTINE LATER'/'C      DC ANALYSIS FOR  ', A8,
                   2  '  USING ROUTINE  ', A8)
0006          123   FORMAT ('0C    THIS IS A  DC ANALYSIS ROUTINE ** ' A8 /
                .  1 6X, 'CALL ', A8, ' ( ', A8, ' REVOLT, RECURT )' )
0007                RETURN
0008                END
```

```
0001                SUBROUTINE SAC  (TYPE, WORD, NWORD, LABEL, ICARD)
                C   SUBROUTINE
                C
                C   A COMPUTATIONAL ROUTINE WILL BE INTENDED TO BE ADDED INTO THE
                C   LANGUAGE, IT WILL BE APEARED IN WORD(4) IN THE ANALYSIS CARD.
                C
0002                DIMENSION LABEL( 9)
0003                DOUBLE PRECISION  FREQ
0004                DOUBLE PRECISION WORD(30), PWORD(30)
0005                DIMENSION MAXLST(10), INTEGR (10), VALUE(20), PRLIST(10, 15)
0006                INTEGER VALUE, PRLIST
0007                DOUBLE PRECISION DUMMY, PRLS
0008                INTEGER  STOPER, TYPE
0009                COMMON/INTBCD/DUMMY
0010                DATA FREQ/'FREQUNCY'/
0011                WRITE (6, 850)  WORD(3), WORD(4)
0012          850   FORMAT('0C     THIS IS A DUMMY SUBROUTINE, IT WILL BE SUBSTITUTED
                   1BY A AC ANALYSIS ROUTINE LATER'/'0C    AC ANALYSIS FOR  ', A8,
                   2  '  USING ROUTINE  ', A8)
0013                DO 10 I=1, NWORD
0014          10    PWORD(I)=WORD(I)
0015          11    CALL FETCH (WORD, PRLIST, MAXLST, NWORD, VALUE, NPRLST,
                   1  STOPER, INTEGR , ICARD, &11,0)
0016                ICARD=ICARD+1
0017                IF  ( WORD(1)  .EQ.  FREQ ) GO TO 20
0018                WRITE (6, 810)
0019          810   FORMAT('0C     ERROR INPUT SETUP**  CARD EXPECTED IS FREQUENCY,
                   1 CARD READS  ')
0020                WRITE  (6, 811) ( WORD(I), I=1, NWORD )
0021          811   FORMAT('0C     ', 10A8)
0022                GO TO 9
0023          20    CALL SCALE (TYPE, WORD, NWORD, LABEL)
0024                WRITE (6, 830)  PWORD(4), PWORD(4), PWORD(3)
0025          830   FORMAT('0C     THIS IS AN AC ANALYSIS ROUTINE ** ' A8 /
                   1 7X, 'CALL ', A8, ' ( ', A8, ' VOLT, CURT )' )
0026          9     RETURN
0027                END
```

```
0001            SUBROUTINE STRT (TYPE, WORD, NWORD, LABEL, ICARD)
          C     SUBROUTINE
          C
          C
0002            DIMENSION LABEL( 9), FREQ(12),IN(2)
0003            DOUBLE PRECISION WORD(30) , PWORD(30)
0004            DOUBLE PRECISION  TIME,K
0005            EQUIVALENCE (IN,K)
0006            COMMON/TIMINT/ FREQ
0007            COMMCN/INTBCD/DUMMY
0008            DIMENSION MAXLST(10), INTEGR (10), VALUE(20), PRLIST(10, 15)
0009            INTEGER VALUE, PRLIST
0010            DOUBLE PRECISICN DUMMY, PRLS
0011            INTEGER  STOPER, TYPE
0012            DATA TIME/'TIME    '/
0013            WRITE (6, 860)  WORD(3), WORD(4)
0014      860   FORMAT('0C     THIS IS A DUMMY SUBROUTINE, IT WILL BE SUBSTITUTED
                1BY A TRANSIENT ANALYSIS ROUTINE LATER'/'0C    THE ANALYSIS IS FOR'
                2 , 2X, A8, ' USING ROUTINE  ', A8)
0015            DO 20 I=1, NWORD
0016      20    PWORD(I) = WORD(I)
0017      22    CALL FETCH (WORD, PRLIST, MAXLST, NWORD, VALUE, NPRLST,
                1  STOPER, INTEGR , ICARD, &22,0)
0018            ICARD=ICARD+1
0019            IF ( WORD(1)  .EQ.  TIME   ) GO TO 21
0020            WRITE (6, 820)
0021      820   FORMAT('0C     **ERROR INPUT SETUP**  CARD EXPECT TRANSIENT, CARD
                1 READS ')
0022            WRITE (6, 821) (WORD(I), I=1, NWORD)
0023      821   FORMAT('0C     ', 10A8)
0024            GOTO 91
0025      21    CALL SCALE (TYPE, WORD, NWORD, LABEL)
0026            WRITE (6, 841)  PWORD(4), PWORD(4), PWORD(3)
0027      841   FORMAT('0C     THIS IS A TRANSIENT ANALYSIS ROUTINE ** ', A8 /
                1 7X, 'CALL ', A8, ' ( ', A8, ' VOLT, CURT )' )
0028            DO 30 I=3,4
0029            N = 8
0030            K = WORD(I)
0031            CALL BCDFPT(ANS,IN,N)
0032            IF(N .EQ. -1) WRITE(6,31) ICARD
0033      31    FORMAT('0ERROR. INVALID VALUE FOR    TIME   ON CARD',I6)
0034      30    FREQ(I -2) = ANS
0035            FREQ(12) = 2.0
0036      91    RETURN
0037            END
```

```
0001            SUBROUTINE SCALE (TYPE, WORD, NWORD, LABEL)
              C SUBROUTINE
              C
              C
0002            DIMENSION LABEL( 9)
0003            DOUBLE PRECISION WORD(30),
               1     EXTERL, END, MODEL, NETWOK, DESCRI, SLASH
0004            INTEGER  STOPER, TYPE
0005            DATA EXTERL  /'EXTERNAL'/,  END/'END     '/, MODEL/'MODEL   '/,
               1      DESCRI/'DESCRIPT'/,  NETWOK/'NETWORK '/, SLASH/'/'/
0006            IF (WORD(3)  .EQ. SLASH)  GO TO 11
0007            GO TO (1, 2), TYPE
0008          1 WRITE (6, 81000)WORD(4), WORD(2), WORD(3), WORD(2),  LABEL(6),
               1   LABEL(6), WORD(3)
0009       81000 FORMAT (/7X, 'IFCOUT = 1'/7X, 'IFREQ = (', A8, '-', A8, ')/', A8,
              1 ' + 1'/7X, 'FREQ(1) = ', A8/7X, 'DO ', I5,' IFCOUT=2, IFREQ'/1X,
              2 I5, ' FREQ(IFCOUT)=( FREQ(IFCOUT-1) + ', A8)
0010            LABEL(6)=LABEL(6)+1
0011            GO TO 999
0012          2 WRITE (6,81002) WORD(4), WORD(2), WORD(3), WORD(2), LABEL(6),
               1  LABEL(6), WORD(3)
0013       81002 FORMAT (/7X, 'ITCOUT = 1'/7X, 'ITIME = (', A8, '-', A8, ')/', A8,
              1 ' + 1'/7X, 'TIME(1) = ', A8/7X, 'DO ', I5,' ITCOUT=2, ITIME'/1X,
              2 I5, ' TIME(ITCOUT) = TIME(ITCOUT-1) +', A8 )
0014            LABEL(6)=LABEL(6)+1
0015            GO TO 999
0016         11 GO TO (3, 4), TYPE
0017          3 LABEL6=LABEL(6)+1
0018            WRITE (6, 81001) WORD(2), LABEL(6), WORD(4), WORD(6), LABEL6,
               1  WORD(4), LABEL(6), LABEL6
0019       81001 FORMAT (/7X, 'IFCOUT = 1' / 7X, 'FREQ(1) = ', A8/1X, I5,' IF (FREQ
              1 (IFCOUT) * ', A8, '.GT. ', A8, ') GO TO ', I5 /7X, 'FREQ(IFCOUT +
              2 1) = FREQ(IFCOUT) * ', A8/7X, 'IFCOUT = IFCOUT + 1'/7X, 'GO TO ',
              3 I5/1X, I5, ' CONTINUE')
0020            LABEL(6)=LABEL6+1
0021            GO TO 999
0022          4 LABEL6=LABEL(6)+1
0023            WRITE (6, 81003) WORD(2), LABEL(6), WORD(4), WORD(6), LABEL6,
               1  WORD(4), LABEL(6), LABEL6
0024       81003 FORMAT (/7X, 'ITCOUT = 1' / 7X, 'TIME(1) = ', A8/1X, I5, ' IF(TIME
              1 (ITCOUT) * ', A8, '.GT. ', A8, ') GO TO ', I5/7X, 'TIME(ITCOUT +
              2 1) = TIME(ITCOUT) * ', A8/7X, 'ITCOUT = ITCOUT + 1'/7X, 'GO TO ',
              3 I5/ 1X, I5, ' CONTINUE')
0025            LABEL(6)=LABEL6+1
0026        999 RETURN
0027            END
```

```
0001            SUBROUTINE NTWORK(MODE, EXTRSV, NOEXEL, NUMEXT, ICARD, STOPER,
               1    NODESV, NTSV, NONTEL, NELMNT, PULVAL,IDSAVE)
          C          **    DECLARATIONS  **
0002            DOUBLE PRECISION PULVAL,          STNMDL(4,30,20), REPEAT(4)
0003            DOUBLE PRECISION SLASH,TEE,FET,SPEC,IMODEL,DUMMY
0004            INTEGER OUTPUT,ELVALU,VALUE,PRLIST,TRINPT,TROUPT,CNTBLK,STOPER
0005            INTEGER  XK, XTRLST, BVALUE
0006            INTEGER EXTNOD,  STNNOD, ICMPNT, ELIST
0007            DIMENSION ELVALU(50,10),VALUE(20),PRLIST(10,15),KNET(2),TRINPT(5,1
               15),TROUPT(5,15),INTEGR(10),NOSTND(6), NOEXND(20), NOSTEL(6)
0008            DOUBLE PRECISION NTSV(7,7), NETWRK,          END, NPORT, NTER, NODE
0009            DOUBLE PRECISION WORD(30), STNDSV(6), DPDUMY,      NETX(2)
0010            DIMENSION MAXLST(10),NOEXEL(20),INPUT(5,10,2),OUTPUT(5,10,2)
0011            DOUBLE PRECISION  BLANK
0012            DOUBLE PRECISION PEE, PRLS, CMPONT(12),MODEL,EXTSAV(20),EXTRSV(20)
0013            DOUBLE PRECISION STNDRD, EXTRNL, EQUAL, DLIST(2,100), TIMES
0014            DIMENSION NODESV(7), NONTEL(7) , NDSV(11), IC(3), PULVAL(3,20,7)
0015            DIMENSION IDSAVE(12),XTRLST(5,30,20), EXTNOD(20,20),BVALUE(20).
0016            DIMENSION  STNNOD(6, 20), ICMPNT(2,12), ELIST(4,100), INZ(2)
0017            COMMON/CAL/ ELIST,ELVALU,XTRLST,INPUT,OUTPUT,TRINPT,TROUPT,DLIST
0018            COMMON /INTBCD/ DUMMY
0019            COMMON/VBREAK/ BVALUE, IBREAK
0020            EQUIVALENCE (CMPONT, ICMPNT), (DUMMY, INZ)
          C          **  DATA STATEMANTS  **
0021            DATA NETWRK/'NETWORK'/, EQUAL/'='/
0022            DATA NPORT, NTER, NODE/'NPORT','NTER','NODE'/,PEE/'P'
               1/,CMPONT/'R','L','C','Z','Y','K','I','V','D','SV','PV','PJ'/
0023            DATA  PRLS/'PRLS    '/, TIMES/'TIMES'/
0024            DATA TEE,FET,STNDRD,EXTRNL/'T       ','FET     ','STANDARD',
               1'EXTERNAL'/,REPEAT/'REPEAT','SERIES','PARALL','CASCAD'/
0025            DATA MODEL/'MODEL   '/,          SLASH/'/       '/,END/'END
               1 '/,SPEC/'SPEC    '/,STNDSV/'DIODE1','DIODE2','TRAN1','TRAN2','FET
               2','FET2'/,NOSTNM/6/,STNMDL/  .2D1, .13D2,                    'L
               3   ',.4D1,.3D1,.7D1,'C       ',.5D1,.13D2,.8D1,'R       ',.145D3,
               4.7D1,.4D1,'L       ',.2D1,.8D1,.4D1,'MU      ',.12D2,2380*-.1D1/,
               5NOSTEL/5,0,0,0,0,0/,BLANK/'       '/,IBLAMK/'    '/
0026            DATA NOSTND/6,0,0,0,0,0/, STNNOD/2,3,13,8,4,7,114*0/
          C          **READ SECTION **
0027            IA=1
0028            INX=1
0029            INM=1
0030            DO 40001 J=1, 7
0031            DO 40001 I=1, 7
0032       40001 NTSV(I, J)=BLANK
0033            DO  40101  I=1, 7
0034            NONTEL(I)=0
0035       40101 NODESV(I)=0
0036            DO 40102 I=1,3
0037       40102 IC(I) = 0
0038            CNTBLK = 0
0039            CONTRL = 0
0040            IB2500 = 0
0041            ICOUNT = 0
0042            IPRT = 0
```

```
0043                ISLSH = 0
0044                NELMNT = 0
0045                NXELMT = 0
0046                NOEXTR = 0
0047                NOPSTL=0
0048                NTRBLK = 0
0049                DO 40105 J = 1, 49
0050                DO 40104 I = 1,10
0051                ELVALU(J,I) = 0
0052          40104 CONTINUE
0053          40105 CONTINUE
                    C     READS A CARD
0054          40000 CALL FETCH(WORD,PRLIST,MAXLST,NWORD,VALUE,NPRLST,STOPER,INTEGR,
                     1  ICARD, &40000,0)
0055                ICARD = ICARD + 1
0056                IF( WORD(1) .NE. NPORT) GO TO 40200
                    C     WORD(1)=NPORT.ENTERED A PORT BLOCK.USE NOPORT FOR WORD(2)-INTEGER
0057                KONTRL = 1
0058                NOPAST = 0
0059                IPRT = IPRT + 1
                    C     NEED TO CONVERT NOPORT TO INTEGER BEFORE  ABLE TO USE 'IDUMY'
0060                IMODEL = WORD(1)
0061                LPRT = 0
0062                ISLSH = 0
0063                NPRT = 0
0064                MPRT = 0
0065                DUMMY = WORD(3)
0066                CALL BCDINT(NOPORT)
0067                IDUMY = 2*NOPORT + 5
0068                DO 40100 I = 5,IDUMY,2
0069                J = I - NOPAST  - 1 + ISLSH
0070                IF( WORD(J) .EQ. PEE ) GO TO 40110
0071                IF ( WORD(J) .NE. SLASH) GO TO 40106
0072                ISLSH = 1
0073                NOPAST = NOPAST + 1
0074                GO TO 40140
0075          40106 IDUM = J/2
0076                WRITE(6,90411) ICARD,IDUM
0077          90411 FORMAT('OMISSING P IN PORT BLOCK DESCRIPTION,LINE',I4,'WORD',I3)
0078                STOPER = 1
0079          40110 IJ =(J + 1 - 3*KONTRL)/2
0080                IF((WORD(J + 1) .EQ. PRLS).AND.(MAXLST(IJ) .EQ. 2)) GO TO 40120
0081                IDUM = J/2
0082                WRITE(6,90412) ICARD,IDUM
0083          90412 FORMAT('OMISSING VALID NODE LIST,LINE ',I4,',WORD ',I2)
0084                STOPER = 1
                    C     WHEN STORING HAVE TO STORE PRLIST(J/3,K) (K=U,MAXLST(J/3)),USING
                    C     WORD (2) .
0085          40120 IF(ISLSH .EQ. 1) GO TO 40130
0086                NPRT = NPRT + 1
0087                LPRT = LPRT + 1
0088                 INPUT(IPRT,NPRT,1) = PRLIST(LPRT,1)
0089                 INPUT(IPRT,NPRT,2) = PRLIST(LPRT,2)
0090                GO TO 40100
```

```
0091        40130 MPRT = MPRT + 1
0092              LPRT = LPRT + 1
0093              OUTPUT(IPRT,MPRT,1) = PRLIST(LPRT,1)
0094              OUTPUT(IPRT,MPRT,2) = PRLIST(LPRT,2)
0095        40140 IF((IDUMY.LE.(NWORD+NOPAST+1-ISLSH)).OR.((J+2).LT.NWORD))GOTO 4010
                  10
0096              NOPAST = NOPAST + J + ISLSH
0097              LPRT = 0
0098              KONTRL = 0
0099              ICARD = ICARD + 1
0100        40150 CALL FETCH(WORD,PRLIST,MAXLST,NWORD,VALUE,NPRLST,STOPER,INTEGR,
                  1   ICARD, &40150,0)
0101        40100 CONTINUE
0102              CNTBLK = 1
0103              GO TO 40000
0104        40200 IF(WORD(1) .NE. NTER) GO TO 40300
0105              NTRBLK = NTRBLK + 1
0106              IMODEL = WORD(1)
0107              CNTBLK = 1
0108              IF((WORD(4) .EQ. PRLS).AND.(WORD(5) .EQ. PRLS)) GO TO 40210
0109              WRITE(6,90413)   ICARD
0110        90413 FORMAT('0INVALID PARAMETER LIST ON WORD #2 OR 3,ON LINE NO. ',I4)
0111              STOPER = 1
0112        40210 I = MAXLST(1) + MAXLST(2)
0113              DUMMY = WORD(3)
0114              CALL BCDINT(NOTERM)
0115              IF ( NOTERM .EQ. I ) GO TO 40220
0116              WRITE(6,90414) ICARD
0117        90414 FORMAT('0NO. OF TERMINALS  NOT = NO. IN PARAMETER LIST-LINE',I4)
0118              STOPER = 1
0119        40220 II = MAXLST(1)
0120              DO 40225 I = 1,II
0121        40225 TRINPT(NTRBLK,I) = PRLIST(1,I)
0122              II = MAXLST(2)
0123              DO 40230 I = 1,II
0124        40230 TROUPT(NTRBLK,I) = PRLIST(2,I)
0125              GO TO 40000
0126        40300 IF(WORD(1)  .NE. NODE) GO TO 40400
            C     ENTERED INTO A NODE BY NODE COMPONENT DESCRIPTION
0127              IF(WORD(2) .EQ. PRLS) GO TO 40305
0128              WRITE(6,90413)   ICARD
0129              STOPER = 1
0130        40305 DO 40310 IPARAM = 1, 12
0131              IF(WORD(3) .EQ. CMPONT(IPARAM)) GO TO 40312
0132        40310 CONTINUE
            C     NOT R,L,C,Z,Y,MU,V,A, NOR D IN1ST WORD-MUST BE T, FET, OR SPEC
0133              IF (MODE  .EQ.  2)  GO TO  40317
0134              IF( (WORD(3) .NE. TEE) .AND. (WORD(3) .NE. FET) ) GO TO 40315
            C     THIS NODE CONTAINS A TRANSISTOR
0135              IF(MAXLST(1) .EQ. 3) GO TO 40320
0136              WRITE(6,90413)   ICARD
0137              STOPER = 1
0138              GO TO 40320
0139        40312 IF ((MODE  .EQ.  1)  .AND.  (IDSAVE(IPARAM)  .LT.  INTEGR(1) ))
```

```
                1    IDSAVE(IPARAM)=INTEGR(1)
0140                 GO TO 40350
0141          40315  IF(WORD(3) .EQ. SPEC) GO TO 40320
0142          90401  FORMAT('0INVALID ELEMENT NAME ON LINE NO. ' ,I4)
0143                 WRITE(6,90401) ICARD
0144                 STOPER = 1
0145                 GO TO 40000
0146          40317  WRITE(6,90424) ICARD
0147          90424  FORMAT('0****. ERROR. HAVE ASKED FOR A NON-LINEAR ELEMENT IN A MOD
                     1EL DESCRIPTION ON CARD #',I6)
0148                 STOPER = 1
0149                 GO TO 40000
0150          40320  XK=6
0151                 IF (WORD(5) .NE. STNDRD .AND. WORD(5) .NE. EXTRNL) XK=5
0152                 IMODEL=WORD(XK)
0153                 DUMMY=WORD(XK+1)
0154                 DO 40321 I=1, 10
0155          40321  NDSV(I)=0
0156                 CALL BCDINT(ISTART)
0157                 J=MAXLST(1)
0158                 DO 40323 I=1, J
0159          40323  NDSV(I)=PRLIST(1,I)
0160                 NDSV(11)=J
0161                 IF (XK .EQ. 6. .AND. WORD(5) .NE. STNDRD) GO TO 40332
0162          40325  DO 40330 IPARAM = 1,NOSTNM
0163                 IF(IMODEL.EQ. STNDSV(IPARAM) ) GO TO 40343
0164          40330  CONTINUE
0165                 WRITE(6,90402) ICARD
0166          90402  FORMAT('0NOT A STANDARD MODEL NAME IN LINE NO. ',I4)
0167                 STOPER = 1
0168                 CONTRL=0
0169                 GO TO 40000
0170          40332  DO 40333 IPARAM = 1,NUMEXT
0171                 IF(EXTRSV(IPARAM).EQ. IMODEL ) GO TO 40334
0172          40333  CONTINUE
0173                 WRITE(6,90404) ICARD
0174          90404  FORMAT('0EXTERNAL MODEL NAME REQUESTED IS NOT DEFINED, LINE',I4)
0175                 STOPER = 1
0176          40334  DO 40335 IPPM  = 1,NOEXTR
0177                 IF(IMODEL.EQ. EXTSAV(IPPM ) ) GO TO 40340
0178          40335  CONTINUE
0179                 NOEXTR = NOEXTR + 1
0180                 EXTSAV(NOEXTR) =IMODEL
0181          40340  CONTRL = 2
0182          43400  NODESV(IJK)=NODESV(IJK)+NOEXND(IPARAM)-NDSV(11)
0183                 IF (NELMNT .EQ. 0) GO TO 40344
0184                 IND = NDSV (11)
0185                 DO 40342 I=1, IND
0186                 DO 40341 J=1, NELMNT
0187                 DO 40341 K=1, 2
0188                 IF (ELIST (K, J) .EQ. PRLIST(1, I) ) GO TO 40342
0189          40341  CONTINUE
0190                 NODESV(IJK)=NODESV(IJK)+1
0191          40342  CONTINUE
```

```
0192                GO TO 40000
0193          40343 CONTRL = 1
0194                GO TO 43400
0195          40344 NODESV(IJK) = NODESV(IJK) + NDSV(11)
0196                GO TO 40000
              C     ENTERING INTO SIMPLE COMPONENT DESCRIPTION - UNLESS A DIODE
0197          40350 IF(MAXLST(1) .EQ. 2) GO TO 40351
0198                WRITE(6,90413)   ICARD
0199                STOPER = 1
0200          40351 IF( WORD(5) .EQ. EQUAL) GO TO 40355
0201                IF( WORD(3) .EQ. CMPONT(9) ) GO TO 40320
0202          40352 IF(MODE .EQ. 2) GO TO 40360
0203                WRITE(6,90403) ICARD
0204          90403 FORMAT('0MISSING ASSIGNMENT STATEMENT ON LINE NO. ',I4)
0205                STOPER = 1
0206                GO TO 40000
0207          40355 NELMNT = NELMNT + 1
0208                ELIST(1,NELMNT)=PRLIST(1,1)
0209                ELIST(2,NELMNT)=PRLIST(1,2)
0210                ELIST(3,NELMNT)=ICMPNT(1, IPARAM)
0211                ELIST(4,NELMNT)=INTEGR(1)
0212                DLIST(1,NELMNT) = NETX(1)
0213                DLIST(2,NELMNT) = 0
0214                IF( IPARAM .GT. 9) GO TO 43561
0215                DO 40356 I = 1,10
0216          40356 ELVALU(NELMNT,I) = VALUE(I)
0217                GO TO 43569
0218          43561 IMV = IPARAM - 9
0219                IC(IMV) = IC(IMV) + 1
0220                ELVALU(NELMNT,1) = -1
0221                ELVALU(NELMNT,2) = IMV
0222                IA = IC(IMV)
0223                ELVALU(NELMNT,3) = IA
0224                IBREAK = 1
0225                DO 43563 I = 1,10
0226          43563 BVALUE(I) = VALUE(I)
0227                DO 43564 I = 11,20
0228          43564 BVALUE(I) = IBLANK
0229                CALL FETCH(WORD,PRLIST,MAXLST,NWORD,VALUE,NPRLST,STOPER,INTEGR,
                  1   ICARD, &40750,1)
0230                DO  43565 I = 1, 7
0231          43565 PULVAL(IMV,IA,I)    = WORD(I)
0232                IF(WORD(5) .EQ.BLANK) PULVAL(IMV,IA,5) = -1
0233          43569 IF (CNTBLK .EQ. 1) DLIST(2,NELMNT) = IMODEL
0234                IF ( (IJK  .NE.  1)  .OR.  (NODESV(1)  .NE.  0 )) GO TO 40357
0235                NODESV(IJK)=NODESV(IJK)+2
0236                GO TO 40000
0237          40357 LLAST=NELMNT-1
0238                DO 40359 K=1, 2
0239                DO 40358 J=1, LLAST
0240                DO 40358 I=1, 2
0241                IF (ELIST(I, J)  .EQ.  ELIST(K, NELMNT))  GO TO 40359
0242          40358 CONTINUE
0243                NODESV(IJK)=NODESV(IJK)+1
```

```
0244        40359 CONTINUE
0245              GO TO 40000
            C     IN 2ND MODE-CHECKING NETWORK DESCRIPTION OF EXTERNAL MODEL
0246        40360 NXELMT = NXELMT + 1
0247              XTRLST(1,NXELMT,NUMEXT) = PRLIST(1,1)
0248              XTRLST(2,NXELMT,NUMEXT) = PRLIST(1,2)
0249              XTRLST(3,NXELMT,NUMEXT) = ICMPNT(1, IPARAM)
0250              XTRLST(4, NXELMT, NUMEXT)=INTEGR(1)
0251              XTRLST(5,NXELMT,NUMEXT) = IPARAM
0252              LLAST=NOEXND(NUMEXT)
0253              DO 40370  K=1, 2
0254              DO 40365  J=1, LLAST
0255              IF (EXTNOD(J, NUMEXT)  .EQ.  XTRLST (K, NXELMT, NUMEXT))  GO
                 1  TO  40370
0256        40365 CONTINUE
0257              NOEXND(NUMEXT)=NOEXND(NUMEXT)+1
0258              I=NOEXND(NUMEXT)
0259              EXTNOD(I, NUMEXT)=XTRLST(K, NXELMT, NUMEXT)
0260        40370 CONTINUE
0261              GO TO 40000
0262        40400 IDMY = CONTRL + 1
0263              GO TO (40402,40427,40410,40415,40428),IDMY
0264        40402 IF(IB2500 .EQ. 0) GO TO 40500
0265              GO TO 40502
            C     THIS MEANS HAVE  A COMPONENT WITH VALUE FOR A MODEL
            C     TRYING TO FIND CORRESPONDING MODEL NO.
            C     CONTRL = 2 THEREFORE EXTERNAL MODEL VALUE ASSIGNMENT
0266        40410 IDNO = IPARAM
0267              CONTRL = 3
0268              LASTEL = NOEXEL(IDNO)
0269        40415 DO 40420 IPARAM = 1, LASTEL
0270              I=XTRLST(5, IPARAM, IDNO)
0271              IF ( CMPONT(I)  .NE.  WORD(1) )  GO TO  40420
0272              IF ( XTRLST(4,IPARAM,IDNO).EQ. INTEGR(1))GO TO 40422
0273        40420 WRITE(6,39)
0274        39    FORMAT(' THEY DO NOT COMPARE')
0275              WRITE(6,90405)
0276        90405 FORMAT('0ELEMENT NAME DOES NOT CORRESPOND TO MODEL DEFINITION')
0277              IF (WORD(1)  .EQ.  END  .AND.  IB2500  .EQ.  0) GO TO 40500
0278              IF (WORD(1)  .EQ.  END  .AND.  IB2500  .EQ.  1) GO TO 40502
0279              STOPER = 1
0280        40422 ICOUNT = ICOUNT + 1
0281              NELMNT = NELMNT + 1
0282              LLAST=NOEXND(ICNO)
0283              DO 40425  J=1, 2
0284              DO 40423  I=1, LLAST
0285              IF (XTRLST(J, IPARAM, IDNO)  .EQ.  EXTNOD(I, IDNO))  GO TO 40424
0286        40423 CONTINUE
0287              STOPER=1
0288              WRITE (6, 90417) ICARD
0289        90417 FORMAT ('** ERROR. SOMETHING WRONG WITH EXTERNAL MODEL SPEC. FOUND
                 1 ON CARD ', I5)
0290        40424 ELIST(J, NELMNT)=ISTART+I-NDSV(11)-1
0291              IF ( I  .LE.  NDSV(11) )  ELIST(J, NELMNT)=NDSV(I)
```

```
0292        40425 CONTINUE
0293              ELIST(3, NELMNT)=XTRLST(3, IPARAM, IDNO)
0294              ELIST(4, NELMNT)=XTRLST(5, IPARAM, IDNO)
0295              DO 40426 I = 1,10
0296        40426 ELVALU(NELMNT,I) = VALUE(I)
0297              DLIST(2, NELMNT)=IMODEL
0298              DLIST(1, NELMNT)=NETX(1)
0299              IF( ICOUNT .NE. NOEXEL(IDNO) ) GO TO 40000
0300              CONTRL = 0
0301              ICOUNT = 0
0302              GO TO 40000
                  C     ENTERING INTO STANDARD MODEL VALUE ASSIGNMENT
0303        40427 IDNO = IPARAM
0304              N = NOSTEL(IDNO)
0305        40428 DO 40430 JPARAM = 1,N
0306              IF( STNMDL(3,JPARAM,IDNO) .NE. WORD(1) ) GO TO 40430
0307              IF ( STNMDL(4,JPARAM,IDNO) .EQ. INTEGR(1))GO TO 40435
0308        40430 CONTINUE
0309              WRITE(6,90405)
0310              IF (WORD(1) .EQ. END .AND. IB2500 .EQ. 0) GO TO 40500
0311              IF(WORD(1) .EQ. END .AND. IB2500 .EQ. 1) GO TO  40502
0312              STOPER = 1
0313        40435 CONTRL = 4
0314              NELMNT = NELMNT + 1
0315              ICOUNT = ICOUNT + 1
0316              LLAST=NOSTND(IDNO)
0317              DO 40444  J=1, 2
0318              DO 40442  I=1, LLAST
0319              IF (STNMDL(J, IPARAM, IDNO) .EQ. STNNOD(I, IDNO) )  GO TO 40443
0320        40442 CONTINUE
0321              STOPER=1
0322              WRITE (6, 90418) ICARD
0323        90418 FORMAT ('** ERROR. SOMETHING WRONG WITH STANDARD MODEL FOUND ON CA
                 1RD ', I5)
0324        40443 ELIST(J, NELMNT)=ISTART+I-NDSV(11)-1
0325              IF ( I .LE. NDSV(11) ) ELIST(J, NELMNT)=NDSV(I)
0326        40444 CONTINUE
0327              ELIST(3, NELMNT)=STNMDL(3, IPARAM, INDO)
0328              ELIST(4, NELMNT)=STNMDL(4, IPARAM, INDO)
0329              DO 40445 I = 1,10
0330        40445 ELVALU(NELMNT,I) = VALUE(I)
0331              DLIST(1, NELMNT)=NETX(1)
0332              DLIST(2, NELMNT)=IMODEL
0333              IF( ICOUNT .NE. NOSTEL(IDNO) ) GO TO 40000
0334              CONTRL = 0
0335              ICOUNT = 0
0336              GO TO 40000
0337        40500 KNET(1) = 0
0338              KNET(2) = 0
0339              ICHECK = 0
0340              ICNTRL = 1
0341              IICNTL = 0
0342              IBCNTL = 0
0343              IB2500 = 1
```

```
            C     ENTERING INTO A NETWORK CHECK
 0344       40502 IF ( WORD(1) .NE. END ) GO TO 40525
 0345             ICHECK = ICHECK + 1
 0346             IF ( WORD(2) .NE. NETWRK) GO TO 40510
 0347             ICHECK = ICHECK - 1
 0348             IF(ICHECK .EQ. KNET(1)) GO TO 40625
 0349             WRITE(6,90406) ICARD
 0350       90406 FORMAT('ONOT SAME NO. OF NETWORK NAMES AND "END" STATEMENTS',I2)
 0351             STOPER = 1
 0352             RETURN
 0353       40510 IF ( WORD(2) .EQ. NETX(1) ) GO TO (40520,40522),MODE
 0354             IF ( WORD(2) .EQ. MODEL ) RETURN
 0355             IF((WORD(2) .EQ. NPORT).OR.(WORD(2) .EQ.NTER)) GO TO 40515
 0356             WRITE(6,90407) ICARD
 0357       90407 FORMAT('OLABEL WITH END STATEMENT DOES NOT CORRESPOND TO NET-NAME
                1"NETWORK", NOR "MODEL" ON CARD ',I4)
 0358             STOPER = 1
 0359             CONTRL = 0
 0360             GO TO 40000
 0361       40515 CNTBLK = 0
 0362             ICHECK = ICHECK - 1
 0363             GO TO 40000
 0364       40520 ICNTRL = 1
 0365             I=KNET(1)
 0366             NCNTEL(I)=NELMNT-NOPSTL
 0367             NOPSTL=NELMNT
 0368             IICNTL = 1
 0369             GO TO 40000
 0370       40522 NOEXEL(NUMEXT) = NXELMT
 0371             CONTRL = 0
 0372             ICNTRL = 1
 0373             NXELMT = 0
 0374             GO TO 40000
            C     ENTERING CHECK FOR REPITION SECTION
 0375       40525 DO 40530 IBCNTL = 1,4
 0376             IF(REPEAT( IBCNTL) .EQ. WORD(1) ) GO TO 40535
 0377       40530 CONTINUE
 0378             GO TO 40600
 0379       40535 IIII = KNET(1)
 0380             DO 40540 IPRM = 1,IIII
 0381             IF(NTSV(IPRM,1).EQ.WORD(2) ) GO TO 40545
 0382       40540 CONTINUE
 0383             WRITE(6,90408) ICARD
 0384       90408 FORMAT('ONOT A VALID NET-NAME AFTER REPITION NAME ON LINE',I4)
 0385             STOPER = 1
 0386             GO TO 40000
 0387       40545 IF(WORD(4) .EQ. TIMES) GO TO 40550
 0388             WRITE(6,90409) ICARD
 0389       90409 FORMAT('OMISSING WORD ''TIMES'' IN REPITION FEATUR, ON LINE ',I4)
 0390             STOPER = 1
            C     HERE NEED TO CALL ROUTINE TO LINK NETWORK
 0391       40550 DUMMY = WORD(3)
 0392             CALL BCDINT(NOTIME)
 0393             WRITE(6,90410) REPEAT(IBCNTL),WORD(3)
```

```
0394          90410 FORMAT('0 A ROUTINE SHPULD BE CALLED FOR REPITION DO 'A8,'-TYPE LI
                   1INKING, 'A8,'TIMES.')
0395                GO TO 40000
              C     ENTERING REGULAR NETWORK DESCRIPTION
0396          40600 NETX(ICNTRL) = WORD(1)
0397                KNET(ICNTRL) = KNET(ICNTRL) + 1
0398                IJK=KNET(1)
0399                IF( ICNTRL .EQ. 1) IN = 0
0400                IN = IN + 1
0401                IIDM = KNET(1)
0402                NTSV(IIDM,IN) = NETX(ICNTRL)
0403                IF(( MODE .NE. 2) .OR. (ICNTRL .NE.1)) GO TO 40602
0404                NUMEXT = NUMEXT + 1
0405                EXTRSV(NUMEXT) = NETX(1)
0406                J=MAXLST(1)
0407                DO 40601  I=1, J
0408          40601 EXTNOD (I, NUMEXT)= PRLIST(1, I)
0409                NOEXND(NUMEXT)=J
0410                GO TO 40000
              C     ICNTRL=1 MEANS LOOKING FOR NEW NETNAME;=2,HAVE ENCOUNTERED NETNAME
              C     IICNTL1 MEANS LOOKING FOR NEW NETNAME:=20 FIRST TIME THRU
0411          40602 IF(ICNTRL .NE. IICNTL ) GO TO 40615
0412                IDUMY = IIDM - 1
0413                DO 40605 IPRM = 1,IDUMY
0414                DO 40605  JPRM = 2,7
0415                IF(NETX(1) .EQ. NTSV(IPRM,JPRM) ) GO TO 40615
0416          40605 CONTINUE
0417                WRITE(6,90416)
0418          90416 FORMAT('ONETWORK NAME NOT PREVIOUSLY USED - ERROR')
0419                STOPER = 1
0420          40615 ICNTRL = 2
0421                GO TO 40000
0422          40625 DO 40630 I=1, NELMNT
0423                IF (DLIST(2,I)  .EQ.  0)  GO TO 40630
0424                IF (DLIST(2,I)  .EQ.  NTER  .OR.  DLIST(2, I)  .EQ.  NPORT)
                   1    GO TO  40630
0425                IDNO=ELIST(4, I)
0426                IDSAVE(IDNO)=IDSAVE(IDNO)+1
0427                ELIST(4, I)=IDSAVE(IDNO)
0428          40630 CONTINUE
0429          40750 RETURN
0430                END
```

```
0001            SUBROUTINE BCDFPT( ANS,BCD,N )
                C            CALLED BY FETCH, SETUP                          7100
                C            NEW 671017 LM.                                  7200
                C       BCDFPT CONVERTS DATA FROM BCD TO FLOATING POINT.      7300
                C       BCD IS AN ARRAY CONTAINING THE N BCD CHARACTERS WHICH 7400
                C       ARE TO BE CONVERTED.                                  7500
                C         I = INDEX OF THE CHARACTER BEING CONVERTED.         7600
                C         J = INDEX CORRESPONDING TO THE DIGIT J-1.           7700
                C         K = 1 WHEN DECODING WHOLE NUMBER PORTION.           7800
                C             2 WHEN DECODING FRACTIONAL PORTION.             7900
                C             3 WHEN DECODING EXPONENT.                       8000
0002            INTEGER DIGIT, E, PLUS, DECPT                                 8100
0003            INTEGER BCD,IB
0004            LOGICAL EXPFLG, DIGFLG, DECFLG, EXSIGN                        8200
0005            DIMENSION BCD(1), KSIGN(3), INTEGR(3), RESULT(3), DIGIT(10)   8300
0006            DATA DIGIT/'0','1','2','3','4','5','6','7','8','9'/          8400
0007            DATA PLUS, MINUS, E, DECPT /'+','-','E','.'/                  8500
0008            DATA IB/'  '/
0009          1 FORMAT(' ENTERED BCDFPT')
0010            WRITE(6,1)
0011            ANS=0.0E1
0012            EXPFLG = .FALSE.                                              8600
0013            DIGFLG = .FALSE.                                              8700
0014            DECFLG = .FALSE.                                              8800
0015            EXSIGN = .FALSE.                                              8900
0016            DO 11 K=1,3                                                   9000
0017            KSIGN(K) = 1                                                  9100
0018            INTEGR(K) = 0                                                 9200
0019         11 CONTINUE                                                      9300
0020            NPLART = 0                                                    9400
0021            K = 1                                                         9500
0022            DO 31 I=1,N                                                   9600
0023            CALL GET( BCD,I,ICHAR )                                       9700
                C*                                                            9800
                C          TEST FOR SIGN, DIGIT, DECIMAL POINT, OR E          9900
                C*                                                           10000
0024            IF ( ICHAR - PLUS ) 13,23,13                                 10100
0025         13 IF ( ICHAR-MINUS ) 14,24,14                                  10200
0026         14 DO 15 J=1,10                                                 10300
0027            IF ( ICHAR-DIGIT(J) ) 15,25,15                               10400
0028         15 CONTINUE                                                     10500
0029            IF(ICHAR .EQ. IB) GO TO 31
0030            IF ( ICHAR-DECPT ) 16,26,16                                  10600
0031         16 IF ( ICHAR-E ) 21,29,21                                      10700
                C*                                                           10800
                C*                                PLUS SIGN                  10900
                C*                                                           11000
0032         23 IF ( DIGFLG ) GO TO 28                                       11100
0033            GO TO 31                                                     11200
                C*                                                           11300
                C*                                MINUS SIGN                 11400
                C*                                                           11500
0034         24 IF ( DIGFLG ) GO TO 27                                       11600
0035            KSIGN(1) = -1                                                11700
```

```
0036            GO TO 31                                                    11800
                C*                                                          11900
                C*                          DIGIT FROM 0 TO 9               12000
                C*                                                          12100
0037          25 INTEGR(K) = 10*INTEGR(K)+J-1                               12200
0038            NPLART = NPLART+K-1                                         12300
0039            DIGFLG = .TRUE.                                             12400
0040            GO TO 31                                                    12500
                C*                                                          12600
                C*                          DECIMAL POINT                   12700
                C*  ONLY ONE DECIMAL POINT PER NUMBER IS ALLOWED.           12800
                C*  DECIMAL POINT IS NOT ALLOWED IN EXPONENT.               12900
                C*                                                          13000
0041          26 IF ( DECFLG ) GO TO 21                                     13100
0042            IF ( EXPFLG ) GO TO 21                                      13200
0043            DECFLG = .TRUE.                                             13300
0044            K = 2                                                       13400
0045            GO TO 31                                                    13500
                C*                                                          13600
                C*                          *E* FOR EXPONENT                13700
                C*  BLANK TIMES TEN ** EXPONENT NOT ALLOWED.                13800
                C*                                                          13900
0046          27 KSIGN(3) = -1                                              14000
0047          28 IF ( EXSIGN ) GO TO 21                                     14100
0048            EXSIGN = .TRUE.                                             14200
0049            GO TO 30                                                    14300
0050          29 IF ( EXPFLG ) GO TO 21                                     14400
0051            IF ( .NOT. DIGFLG ) GO TO 21                                14500
0052          30 EXPFLG = .TRUE.                                            14600
0053            K = 3                                                       14700
0054            NPLASV = NPLART                                             14800
0055          31 CONTINUE                                                   14900
                C*                                                          15000
                C     THE NUMBER HAS BEEN SEPARATED INTO INTEGER, FRACTION, AND 15100
                C     EXPONENT PARTS.  COMBINE THEM TO FORM THE NUMBER IN FLOATING 15200
                C     POINT.                                                15300
                C                                                           15400
0056            IF ( EXPFLG ) GO TO 32                                      15500
0057            EXPON = 1.                                                  15600
0058            GO TO 35                                                    15700
                C*                                                          15800
                C     CALCULATE EXPONENT.  AN EXPONENT MAY BE ONLY TWO DIGITS LONG 15900
                C     AND LESS THAN 38 IN MAGNITUDE.                        16000
                C*                                                          16100
0059          32 IF ( NPLART-NPLASV-4 ) 33,33,21                            16200
0060          33 IEXPON = INTEGR(3)*KSIGN(3)                                16300
0061            IF ( IABS( IEXPON ) - 37 ) 34,34,21                         16400
0062          34 EXPON = 10.**IEXPON                                        16500
0063            NPLART = NPLASV                                             16600
                C*                                                          16700
                C*                          CALCULATE MANTISSA              16800
                C*                                                          16900
0064          35 RTSHFT = 10.**NPLART                                       17000
0065            RESULT(1) = FLOAT( INTEGR(1)*KSIGN(1) )                     17100
0066            RESULT(2) = FLOAT( INTEGR(2)*KSIGN(1) ) / RTSHFT            17200
0067            ANS = (RESULT(1) + RESULT(2) )*EXPON                        17300
0068          41 RETURN                                                     17400
                C*                                                          17500
                C     ILLEGAL CHARACTER OR BAD SYNTAX.                      17600
                C*                                                          17700
0069          21 N = -1                                                     17800
0070            GO TO 41                                                    17900
0071            END                                                         18000
```

```
0001            SUBROUTINE SPRT (ANALYS, WORD, NWORD, NTNAME, LABEL, ICARD, NTSV
               1    ,  NODESV, NONTEL)
                C     SUBROUTINE
                C
                C
0002            DIMENSION LABEL( 9), NODESV(7), NONTEL(7), IPCURR(2,100),IPVOLT(2,
               A 100) , ELIST(4,100)
0003            DOUBLE PRECISION  WORD(30), NTNAME, CURT, VOLT, ANALYS
0004            DOUBLE PRECISION  NTSV(7,7)
0005            INTEGER  BRANCH, ELIST
0006            COMMON/PRINTE/ IPCURR, IPVOLT, NELMNT
0007            COMMON /CAL/ ELIST
0008            DATA CURT/'CURRENT '/, VOLT/'VOLTAGE '/,LK,LV,LSV,LPV,LVI/'K','V',
               8'SV','PV','I'/
0009            DO 10 I = 1,2
0010            DO 10 J = 1,100
0011            IPCURR(I,J) = 0
0012       10   IPVOLT(I,J) = 0
0013            DO  400  I=2, NWORD
0014            WRITE(6,10000) I,WORD(I), NWORD
0015       10000 FORMAT('0WORD(',I3,') =',A9,' OF POSSILBE =',I5)
0016            IF (WORD (I)  .EQ. ' CURT )  GO TO  600
0017            WRITE(6,10001)
0018       10001 FORMAT('NOT CURRENT')
0019            IF (WORD (I)  .EQ.  VOLT )  GO TO  500
0020            WRITE (6, 810)  ICARD, WORD(I)
0021       810  FORMAT (1X, 'C     ****ERROR OUTPUT VARIABLE NAME IN CARD NO. ',
               1  I4, 'THE VARIABLE READS ', A8, '****')
0022            GO TO 400
0023       500  DO  510  J=1, 7
0024            IF ( NTNAME  .EQ.  NTSV(J, 1) )  NNODE=NODESV(J)
0025       510  CONTINUE
0026            WRITE (6, 815)  NTNAME, NNODE
0027       815  FORMAT (20X, '**NUMBER OF NODE IN NETWORK ', A8, '=', I5)
0028            IF (NNODE  .LE.  1)  GO TO 400
0029            WRITE (6, 820)  LABEL(8), LABEL(8), ANALYS, NTNAME
0030       820  FORMAT (7X, 'WRITE (6, ', I5, ' )'/1X, I5, 1X, 'FORMAT (///10X, ',
               1  A8, ' ANALYSIS OUTPUT FOR ', A8, ' )')
0031            LABEL(8)=LABEL(8)+1
0032            WRITE (6, 825)  LABEL(8), NNODE, LABEL(8)
0033       825  FORMAT (7X, 'WRITE (6, ', I5, ' )(VOLT (I), I=1, ', I5, ' )'/
               1  1X, I5, 1X, 'FORMAT.....')
0034            LABEL(8)=LABEL(8)+1
0035            L = 0
0036            DO 41 M=1,NELMNT
0037            IF(ELIST(3,M) .EQ. LK .OR. ELIST(3,M) .EQ. LV) GO TO 41
0038            L = L + 1
0039            IPVOLT(1,L) =ELIST(3,M)
0040            IPVOLT(2,L) =ELIST(4,M)
0041       41   CONTINUE
0042            IPVOLT(2,100) = L
0043            GO TO 400
0044       600  DO  610  J=1, 7
0045            IF ( NTNAME  .EQ.  NTSV(J, 1) )  BRANCH=NONTEL(J)
0046       610  CONTINUE
0047            IF (BRANCH .LE.  1)  GO TO 400
0048            WRITE (6, 855)  NTNAME, BRANCH
0049       855  FORMAT (20X, '**NUMBER OF BRANCHES IN NETWORK ', A8, '=', I5)
0050            WRITE (6, 820)  LABEL(8), LABEL(8), ANALYS, NTNAME
0051            LABEL(8)=LABEL(8)+1
0052            WRITE (6, 825)  LABEL(8), BRANCH, LABEL(8)
0053            LABEL(8)=LABEL(8)+1
0054            L = 0
0055            DO 40 M = 1,NELMNT
0056            IF(ELIST(3,M) .EQ. LK .OR. ELIST(3,M) .EQ. LV .OR. ELIST(3,M) .EQ.
               1 LSV .OR. ELIST(3,M) .EQ. LPV) GO TO 40
0057            L = L + 1
0058            IPCURR(1,L) =ELIST(3,M)
0059            IPCURR(2,L) =ELIST(4,M)
0060       40   CONTINUE
0061            IPCURR(2,100) = L
0062       400  CONTINUE
0063            RETURN
0064            END
```

```
0001          SUBROUTINE CIRINT(ELIST,NELMNT,ELVALU,IDSAVE,FREQ,IPCURR,IPVOLT,
             A NCNCDE,PULVAL)
0002          DOUBLE PRECISION PULVAL(3,20,7), DP
0003          DIMENSION NI(2),IG(2), SIMPJ(5)                                    7
0004          INTEGER ELIST, ELVALU, VEE
0005          DIMENSION IDSAVE(12),FREQ(12),IPCURR(2,100),IPVOLT(2,100),IELTYP(1
             A0),ISAVE(10), ELIST(4,100), ELVALU(50,10),IDP(2)
0006          EQUIVALENCE (WORD,NWORD), (ILPDS,LPDS), (ILDS,LDS),(DP,IDP)
0007          LOGICAL ENDFLG, HOLD, HOLD2, MODLIN                                298400
0008          INTEGER TRLIST, DILIST, FETLST, TDLIST, TITLE, PEAKGM              298500
0009          INTEGER DIDICT, FETDCT, TDDICT, TRDICT                             298600
0010          DIMENSION WORD(1),NWORD(1),MNSAVE(12)                              298700
0011          DIMENSION NJFT(1), LOCBB(1), NT1(1)                                298800
0012          COMMON  WORD                                                       298900
0013          COMMON  N1    , N5    , N6    , LPDS  , LDS   , DUMMY(12)          299000
0014          COMMON  NTOT  , NJV   , NJSV  , NJPV  , NJCL  , NJC                299100
0015          COMMON  NJR   , NJL   , NJPJ  , NJM   , NJT   , NJTL               299200
0016          COMMON  NJD   , NJDL  , LINTVL, LGMDOT, LPLTVL, PEAKGM             299300
0017          COMMON  ENDFLG, N2    , N3    , NEL   , NODES , NBRNCH             299400
0018          COMMON  NCHORD, NJB   , NYSIZE, NXSIZE, NZSIZE, LELNAM             299500
0019          COMMON  LELVAL, LAMAT , NJTHET, NJDELT, NJEPS , NJZETA             299600
0020          COMMON  NJALPH, NJBETA, NJGAM , NJPHI , LENA  , LEVA               299700
0021          COMMON  LENB  , LEVB  , LENG  , LEVG  , LENP  , LEVP               299800
0022          COMMON  LENT  , LEVT  , LEND  , LEVD  , LENE  , LEVE               299900
0023          COMMON  LENZ  , LEVZ  , LTRIP , LMTPAR, LTRPAR, LTDPAR             300000
0024          COMMON  LDPAR , LSVOLT, LPVOLT, LPCURR, LCLEAK, LTRLST             300100
0025          COMMON  LDILST, LZGG  , LZGD  , LZDG  , LZDD  , LSDGP              300200
0026          COMMON  LSDDT , LSGP  , LSGT  , LSDP  , LSDT  , LQGG               300300
0027          COMMON  LPDD  , LSCR  , LFPDD , LFQGG , LFQBB , LFPEE              300400
0028          COMMON  NVSAVE, LVHOLD, LVALIC, LVALIP, LFSDGP, LFSDDT             300500
0029          COMMON  LFSGP , LFSGT , NPCELL, LPLT  , LLZZ  , LCZGD              300600
0030          COMMON  LCZPD , LCEG  , LCED  , LCEP  , LCET  , LFCZGD             300700
0031          COMMON  N4    , LFCEG , LFCED , LFCEP , LFCET , N7                 300800
0032          COMMON  NJMT  , NJMTL , NJTD  , NJTDL , INDDEV, MINT               300900
0033          COMMON  LSPC  , TMAX  , TNEXT , NRGIP , NTIP  , LCC                301000
0034          COMMON  LRATNG, LMAXMS, NMAXMS, NLC   , NTPV  , JXX                301100
0035          COMMON  NTPJ  , JIPP  , LCHNGE, KSPCAL, NVSRCE, NSWV               301200
0036          COMMON  NPV   , NVDS  , NVE   , NVZ   , NJSRCE, NDSWV              301300
0037          COMMON  NDPV  , NDJSRC, NSL   , NSRCE , NSX   , NSFT               301400
0038          COMMON  NVL   , NVD   , NVX   , LFLAGX, NVWORK, NDVC               301500
0039          COMMON  LPBB  , LTEMP , NPPRNT, NPPLOT, NALZ  , NNV                301600
0040          COMMON  NPRNT , NIZ   , NIBCD , NPAGE , NPNAME, NCUR               301700
0041          COMMON  NELP  , NSPR  , NELG  , NSG   , NJPULS, LLINK4             301800
0042          COMMON  NOIPP , LF    , LR    , LRSTAB, NMODLS , NJX               301900
0043          COMMON  LSCPM , NXC   , NEQ   , NGDL  , NGD   , LTRSET             302000
0044          COMMON  INDMDT, IPLOTI, NPLOTS, KPRNT , NPNT  , DTCUR              302100
0045          COMMON  LAX   , LEN   , LPHI  , LTHA  , LXIS  , LTC1               302200
0046          COMMON  LMDL  , LLOCB , LLOCT , LNBN  , LNTC  , LXI                302300
0047          COMMON  XLISTX(14784), NELTYP(500), JNEL(500)                      302400
0048          EQUIVALENCE (WORD(1),NWORD(1),NZERO)                               302500
0049          COMMON /EXTCOM/  NJFT,  NJZD, NJFTL, NJZDL, LFTPAR, LZDPAR         302600
             1      , LFTLST, LZDLST                                            302700
0050          COMMON/DICTCM/FETDCT(21),TRDICT(38),DIDICT(20),TDDICT(19),         302800
             1              MTDICT(47),ZDDICT                                   302900
```

```
0051          COMMON /MDLNAM/ IRLIST(25), DILIST(25), MTRLST(5), TDLIST(25)      303000
             1       , FETLST(25), ZDLIST(25)                                    303100
0052          COMMON /MISC  / KET,HOLD,LREST,NPEAK,TIME,IRATE,LSAVE,L,NBIAS      303200
             1    , JNODES(6), JPRINT(6), JPLOT(6), PLTINT, HOLD2, JFUNCT        303300
0053          COMMON /SCRTCH/ SKIP(99),KOUNT,ZNAME(100),IDUM1(400)                  7
0054          COMMON /MTXEQ1/ LOCBB, LOCTCB, THETMT(10), XISMT(10), NBB(10),     303500
             1                NTCB(10)                                           303600
0055          COMMON /VINDEX/    NT1,    NT2,    ND1,    ND2,    NZD1,    NZD2  303700
             1       , NET1,   NET2,   NMT1,   NMT2,   NTD1,   NTD2             303800
0056          COMMON /CCMGIC/ NGUESS, NGTRUE, GUESIC(70)                         303900
0057          COMMON /TITLE / TITLE(20)                                          304000
0058          DATA IPP,IPPS,NBLNK,NQX00 / 'IPP','IPPS',' ','S'/                  304100
0059          DATA NQX01, NQX02 / 'P', 'J' /                                     304200
            C*************************************************************      304300
            C                                                                    304400
0060          DATA IB/'    '/,BL/'    '/, SIMPJ/0.0, 0.0, 10.0, 0.0, 10.0/
0061          DATA PRIN/'PRIN'/,TEE/'T'/,IELTYP/ 'K','V','SV',
             A 'PV', 'LC', 'C', 'R', 'L', 'I','PJ'/
0062          DATA IEE/'I   '/,VEE/'V   '/,   EXEC,UTE/'EXEC','UTE'/
0063          WRITE(6,1)
0064       1  FORMAT(' ENTERED CIRINT')
0065          CALL ZYXSPM
0066          CALL KLOCK1(1.E7)
0067          ILDS = 15000
0068          ILPDS = 217
0069          KTEST = 0
0070          IVLTNO = IPVOLT(2,100)
0071          ICURNO  = IPCURR(2,100)
0072          DO 300 NLMT = 1,NELMNT
0073          LDSTEM = 0
0074          NWORD(LDS) = ILPDS
0075          IF( ELIST(3,NLMT) .EQ. IELTYP(1) ) GO TO 102
0076          NWORD(LDS - 1) = ELIST(1,NLMT)
0077          NWORD(LDS - 2) = ELIST(2,NLMT)
0078          IF( ELVALU(NLMT,1) .EQ. -1) GO TO 110
0079          GO TO 101
0080       102  NWORD(ILDS - 1) = LPDS + 2
            C   THIS IS A "K" ELEMENT
0081          NWORD(ILDS - 2) = LPDS + 4
0082          KTEST = -2
0083       101  DO 100 I = 1,10
0084       100  ISAVE(I) = ELVALU(NLMT,I)
0085          WRITE (6, 93) NLMT, (ISAVE(I), I=1, 10)
0086       93   FORMAT (' ELVALU #', I2, ' IS ', 10A4)
0087          N=40
0088          CALL BCDFPT( ANS,ISAVE, N)
0089          IF (N  .EQ.  -1)  WRITE (6, 92) NLMT
0090       92   FORMAT (' C      ** ERROR **  INVALID VALUE ASSIGNED TO ELEMENT NU
             1 ', I4)
0091          WRITE(6,95) ANS, ANS
0092       95   FORMAT(' ANS IS ',F8.3,' OR ',E12.5)
0093          WRITE(6,94) IANS
0094       94   FORMAT(' IANS IS ',I8)
0095          WORD(ILDS - 3) =  ANS
```

```
0096            IF( ELIST(3,NLMT) .NE. IELTYP(9) ) GO TO 120
                C   SIMULATING A FIXED CURRENT SOURCE ("I"), WITH A PULSED CURRENT SOURC
0097            LDSTEM = LDSTEM + 1
0098            WORD(LDS-3 - LDSTEM) = ANS
0099            DO 103 I = 1, 5
0100            LDSTEM = LDSTEM + 1
0101        103 WORD(LDS-3-LDSTEM) = SIMPJ(I)
0102            ELVALU(NLMT,1) = -1
0103            ELVALU(NLMT,2) = 3
0104            GO TO 120
0105        105 KTEST = KTEST + 1
                C  CREATING THE CORRESPONDING INDUCTORS FOR THE COUPLING EFFEST ("K")
0106            LPDS = LPDS + 2
0107            IA = IELTYP(8)
0108            ID = ELIST(KTEST + 2,NLMT)
0109            GO TO 170
0110        110 DO 115 I =1, 7
0111            K = ELVALU(NLMT,2)
0112            J = ELVALU(NLMT,3)
0113            DP = PULVAL( K, J, I)
0114            IF( DP .EQ. -1) GO TO 119
0115            N = 8
0116            CALL BCDFPT(ANS,IDP, N)
0117            IF (N  .EQ.  -1)  WRITE (6, 92) NLMT
0118            WORD(LDS-3-LDSTEM) = ANS
0119        115 LDSTEM = LDSTEM + 1
0120        119 LDSTEM = LDSTEM - 1
0121        120 WORD(ILDS - 4 - LDSTEM) = 1.0
0122            IST = 2
0123            IA = ELIST(3,NLMT)
0124            DO 130 I = 1,9
0125            J = 10 - I
0126            IF(ELIST(3,NLMT).EQ. IELTYP(J) ) GO TO 135
0127        130 CONTINUE
                C   ASSUMIING ITS "PJ"
0128            J = 9
0129        135 NWORD(15000 +NLMT) = 10*(J - 1)
0130            NWORD(15500 +NLMT) = LDS
0131            ID = ELIST(4,NLMT)
0132            DO 150 I = 1,ICURNO
0133            IF(IA .NE. IPCURR(1,I) .OR.  ID .NE. IPCURR(2,I)) GO TO 150
0134            IPCURR(1,I) = LPDS
0135            IF( ELVALU(NLMT,1) .EQ. -1 ) IPCURR(2,I) = -ELVALU(NLMT,2)
0136        150 CONTINUE
0137            DO 160 I = 1,IVLTNO
0138            IF(IA .NE. IPVOLT(1,I) .OR.  ID .NE. IPVOLT(2,I))  GO TO 160
0139            IF( ELVALU(NLMT,1) .EQ. -1 ) IPVOLT(2,I) = -ELVALU(NLMT,2)
0140            IPVOLT(1,I) = LPDS
0141        160 CONTINUE
0142            IF( IA .NE. IELTYP(9) ) GO TO 165
0143            IA = IELTYP(10)
0144            ID = ID + IDSAVE(12)
0145        165 CALL GET(IA,2,ITEST)
0146            IF(ITEST .NE. IB) IST =3
```

```
0147            WRITE (6, 8) ID
0148       8    FORMAT ( ' C     *****ENTR INTBCD FROM HERE ID= ', I5)
0149      170   CALL INTBC (ID,ICHAR,L)
0150            WRITE (6, 9)
0151       9    FORMAT ( ' C***** RETURNED FROM INTBCD')
0152            DO 200 I = 1,L
0153            J = IST + I - 1
0154            CALL GET( ICHAR,I,ITEST)
0155            WRITE(6,96) ICHAR,ITEST
0156       96   FORMAT(' ICHAR IS ',A4,' ITEST IS ',A4)
0157            CALL PUT(IA,J,ITEST)
0158            IF( ITEST .EQ. IB) GO TO 250
0159      200   CONTINUE
0160      250   NWORD(ILPDS) = IA
0161            WORD(ILPDS + 1 ) = BL
0162            IF(KTEST .LT. 0) GO TO 105
0163            ILDS = ILDS - 5 - LDSTEM
0164            ILPDS = ILPDS + 2
0165      300   CONTINUE
0166            WRITE(6,51)((IPCURR(J,I),J=1,2),I=1,ICURNO)
0167       51   FORMAT(' IPCURR''S LOCATION IN WORD IS'/5(/10(I7,I4) )  )
0168            WRITE(6,52)((IPVOLT(J,I),J=1,2),I=1,IVLTNO)
0169       52   FORMAT(' IPVOLT''S LOCATION IN WORD IS'/5(/10(I7,I4) )  )
             C    START BY ZEROING PARAMETERS AS REQUIRED.                  304700
0170            INDDEV = 0                                                7  360
0171            LIST = INDDEV                                             7  360
0172            DO 502 I = 1,8                                            304900
0173      502   NJFT(I) = 0                                               305000
0174            NGUESS = 0                                                305100
0175            NGTRUE = 0                                                305200
0176            DO 501 I = 7,216
0177      501   NWORD(I) = 0                                              305400
0178            DO 503 I = ILPDS,300
0179      503   WORD(I) = 0
0180            INDDEV =LIST                                              305450
0181            DO 504 I=1,20                                             305500
0182      504   TITLE(I) = NBLNK                                          305600
0183            ENDFLG = .TRUE.                                           305700
0184            MODLIN = .FALSE.                                          305800
0185            HOLD2 = .FALSE.                                           305900
0186            HOLD = .FALSE.                                            306000
0187            JFUNCT = 0
0188            LSAVE = 1                                                 306100
0189            LREST = 1                                                 306200
0190            INDSEG = 1                                                306300
0191            PLTINT = 0                                                306400
0192            INDMDT=1                                                  306500
0193            LCHNGE = 0                                                306600
0194            NPEAK = 0                                                 306700
0195            NBIAS = 0                                                 306800
0196            NBAD = 0                                                  306900
0197            JNODES(1) = 0                                             307000
             C  * LPDS = RUNNING INDEX FOR PERMANENT STORAGE.             307100
             C    LDS = RUNNING INDEX FOR ERASABLE INPUT STORAGE.         307200
```

```
      C   * ASSIGN TAPE UNITS.                                                307500
      C         N5 AND N6 ARE THE SYSTEM INPUT AND OUTPUT UNITS.              307600
      C         N1 WILL HAVE MODEL PARAMETERS WHICH ARE READ FROM THE         307700
      C            INPUT STREAM.                                              307800
      C         N2 IS THE PLOT TAPE.                                          307900
      C         N3 IS THE SAVE TAPE                                           308000
      C         N8 IS THE RESTART TAPE                                        308100
      C         N4 IS THE DEVICE PARAMETER LIBRARY (MAY BE CHANGED AT &2601). 308200
      C         N7 IS A CIRCUS SCRATCH TAPE (USED IN LINK4).                  308300
0198        NZERC=0                                                           308400
0199        N1 = 11                                                           308500
0200        N2 = 12                                                           308600
      C                                                                       308700
0201        N3 = 3                                                            308800
0202        N4 = 4                                                            308900
0203        N5=5                                                              309000
0204        N6=6                                                              309100
0205        N7 = 8                                                            309200
0206        REWIND N7                                                         309400
      C*****************************************************************     309500
0207        NTOT     = NONODE - 1
      C IDSAVE 1=R,2=L,3=C,4=Z,5=Y,6=K,7=2,8=V,9=D,10=SV,11=PV,12=PJ
0208        NJV      = IDSAVE( 8)
0209        NJSV     = IDSAVE(10)
0210        NJPV     = IDSAVE(11)
0211        NJC      = IDSAVE( 3)
0212        NJR      = IDSAVE( 1)
0213        NJL      = IDSAVE( 2)
0214        NJPJ     = IDSAVE( 7) + IDSAVE(12)
0215        NJM      = IDSAVE( 6)
0216        NEL      = NELMNT
0217        ZZ = FREQ(12)
0218        IFREQ = INT(ZZ)
0219        WRITE(6,70) ICURNO,IVLTNO,IFREQ
0220     70 FORMAT(' IPCURR =', I8,' IVLTNO =',I8,' IFREQ = ',I8)
0221        WORD(LPDS) = IFREQ + 1
0222        LINTVL = LPDS
0223        TMAX = FREQ(IFREQ)
0224        NWORD(LPDS)= FREQ(12) + 1
0225        DO 320 I = 1,IFREQ
0226     320 WORD(LPDS + I) = FREQ(I)
0227        LPDS = LPDS + IFREQ + 1
0228        KPRNT = ICURNO + IVLTNO
0229        WORD(LPDS) = PRIN
0230        WORD(LPDS+1)= TEE
0231        LPDS = LPDS + 2
0232        NPNAME     = LPDS + 2*KPRNT
0233        LI = LPDS + 2*KPRNT - 1
0234        IF(ICURNO .EQ. 0) GO TO 350
0235        DO 340 I = 1,ICURNO
0236        IG(1) = IEE                                                          7
0237        IG(2) = IB                                                           7
0238        J = 1
0239        IA = IPCURR(1,I)
```

```
0240            NI(1) =NWORD(IA)
0241            NI(2) =NWORD(IA + 1)
0242            IF( IPCURR(2,I) .NE. -3) GO TO 325
0243            IG(1) = NI(1)
0244            GO TO 335
0245      325   DO 330 K = 1,7
0246            CALL GET(NI,K,ITEST)
0247            J = J + 1
0248      330   CALL PUT(IG,J,ITEST)
0249      335   NWORD(LPDS)= IG(1)
0250            WRITE(6,81) ILPDS,IG(1)
0251      81    FORMAT(' WORD(',I6,')=',A4)
0252            WRITE(6,81) ILPDS,IG(2)
0253            NWORD(LPDS+ 1) = IG(2)
0254            LI = LI + 1
0255            NWORD(LI)= LPDS
0256      340   LPDS = LPDS + 2
0257      350   IF(IVLTNO .EQ. 0) GO TC 400
0258            DO 380 I = 1,IVLTNO
0259            J = 1
0260            IG(1) = VEE
0261            IG(2) = IB                                              7
0262            IA = IPVOLT(1,I)
0263            NI(1) =NWORD(IA)
0264            NI(2) =NWORD(IA + 1)
0265            IF( IPVOLT(2,I) .GE. 0 .OR. IPVOLT(2,I) .EQ. -3) GO TO 355
0266            IG(1) = NI(1)
0267            GO TO  365
0268      355   DO 360 K = 1,7
0269            CALL GET(NI,K,ITEST)
0270            J = J + 1
0271      360   CALL PUT(IG,J,ITEST)
0272      365   NWORD(LPDS)= IG(1)
0273            NWORD(LPDS+ 1) = IG(2)
0274            LI = LI + 1
0275            NWORD(LI)= LPDS
0276      380   LPDS = LPDS + 2
0277      400   IF(LI .GT. LPDS)LPDS = LI + 1
0278            WORD(LPDS) = EXEC
0279            WORD(LPDS + 1) = UTE
0280            NWORD(LDS)= LPDS
0281            J = 0
0282            IEND = LPDS - 1
0283            DO 420 I = NPNAME,IEND
0284            J = J + 1
0285      420   NWORD(LDS- J ) =NWORD(I)
0286            CALL LINK2
0287            CALL MAIN2
0288            CALL ENDJOB
0289            RETURN
0290            END
```

```
0001            SUBROUTINE INTBC (NUM,ICHAR,L)
0002            INTEGER DIFF, R
0003            DIMENSION IN(11)
0004            DATA IN/'0','1','2','3','4','5','6','7','8','9','  '/
0005            WRITE(6,1) NUM
0006       1    FORMAT(' ENTERED INTBCD NUM= ', I5)
0007            ICHAR = IN(11)
0008            IC = 0
0009            L = 0
0010            INM = NUM
0011            K = 1
0012            DO 20 I = 1,4
0013            DIFF = NUM*10/10000
0014            NUM = 10*NUM - 10000*DIFF
0015            DO 10 J = 1,10
0016            R = J - 1
0017            IF ( R .EQ. DIFF ) GO TO 15
0018       10   CONTINUE
0019            WRITE(6,11) INM,K
0020       11   FORMAT('***   CANNOT CHANGE NUMBER',I3,'DIGIT #',I2)
0021            RETURN
0022       15   IF ( (R .EQ.0 ).AND.(IC .EQ. 0).AND.(I .LT. 4) ) GO TO 20
0023            IC = 1
0024            INN = IN(J)
0025            L = L + 1
0026            CALL PUT(ICHAR,K,INN)
0027            K = K + 1
0028       20   CONTINUE
0029            NUM = INM
0030            RETURN
0031            END
```

```
0001            SUBROUTINE BCDFPT( ANS,BCD,N )
                C               CALLED BY FETCH, SETUP                          7100
                C               NEW 671017 LM.                                  7200
                C          BCDFPT CONVERTS DATA FROM BCD TO FLOATING POINT.      7300
                C          BCD IS AN ARRAY CONTAINING THE N BCD CHARACTERS WHICH 7400
                C          ARE TO BE CONVERTED.                                 7500
                C            I = INDEX OF THE CHARACTER BEING CONVERTED.        7600
                C            J = INDEX CORRESPONDING TO THE DIGIT J-1.          7700
                C            K = 1 WHEN DECODING WHOLE NUMBER PORTION.          7800
                C                2 WHEN DECODING FRACTIONAL PORTION.            7900
                C                3 WHEN DECODING EXPONENT.                      8000
0002            INTEGER DIGIT, E, PLUS, DECPT                                   8100
0003            INTEGER BCD,IB
0004            LOGICAL EXPFLG, DIGFLG, DECFLG, EXSIGN                          8200
0005            DIMENSION BCD(1), KSIGN(3), INTEGR(3), RESULT(3), DIGIT(10)     8300
0006            DATA DIGIT/'0','1','2','3','4','5','6','7','8','9'/            8400
0007            DATA PLUS, MINUS, E, DECPT /'+','-','E','.'/                    8500
0008            DATA IB/'  '/
0009          1 FORMAT(' ENTERED BCDFPT')
0010            WRITE(6,1)
0011            ANS=0.0E1
0012            EXPFLG = .FALSE.                                                8600
0013            DIGFLG = .FALSE.                                                8700
0014            DECFLG = .FALSE.                                                8800
0015            EXSIGN = .FALSE.                                                8900
0016            DO 11 K=1,3                                                     9000
0017            KSIGN(K) = 1                                                    9100
0018            INTEGR(K) = 0                                                   9200
0019         11 CONTINUE                                                        9300
0020            NPLART = 0                                                      9400
0021            K = 1                                                           9500
0022            DO 31 I=1,N                                                     9600
0023            CALL GET( BCD,I,ICHAR )                                         9700
                C*                                                              9800
                C       TEST FOR SIGN, DIGIT, DECIMAL POINT, OR E               9900
                C*                                                             10000
0024            IF ( ICHAR - PLUS ) 13,23,13                                   10100
0025         13 IF ( ICHAR-MINUS ) 14,24,14                                    10200
0026         14 DO 15 J=1,10                                                   10300
0027            IF ( ICHAR-DIGIT(J) ) 15,25,15                                 10400
0028         15 CONTINUE                                                       10500
0029            IF(ICHAR .EQ. IB) GO TO 31
0030            IF ( ICHAR-DECPT ) 16,26,16                                    10600
0031         16 IF ( ICHAR-E ) 21,29,21                                        10700
                C*                                                             10800
                C*                          PLUS SIGN                          10900
                C*                                                             11000
0032         23 IF ( DIGFLG ) GO TO 28                                         11100
0033            GO TO 31                                                       11200
                C*                                                             11300
                C*                          MINUS SIGN                         11400
                C*                                                             11500
0034         24 IF ( DIGFLG ) GO TO 27                                         11600
0035            KSIGN(1) = -1                                                  11700
```

```
0036            GO TO 31                                                        11800
                C*                                                              11900
                C*                                  DIGIT FROM 0 TO 9           12000
                C*                                                              12100
0037          25 INTEGR(K) = 10*INTEGR(K)+J-1                                   12200
0038            NPLART = NPLART+K-1                                             12300
0039            DIGFLG = .TRUE.                                                 12400
0040            GO TO 31                                                        12500
                C*                                                              12600
                C*                                  DECIMAL POINT               12700
                C*    ONLY ONE DECIMAL POINT PER NUMBER IS ALLOWED.             12800
                C*    DECIMAL POINT IS NOT ALLOWED IN EXPONENT.                 12900
                C*                                                              13000
0041          26 IF ( DECFLG ) GO TO 21                                         13100
0042            IF ( EXPFLG ) GO TO 21                                          13200
0043            DECFLG = .TRUE.                                                 13300
0044            K = 2                                                           13400
0045            GO TO 31                                                        13500
                C*                                                              13600
                C*                                  *E* FOR EXPONENT            13700
                C*    BLANK TIMES TEN ** EXPONENT NOT ALLOWED.                  13800
                C*                                                              13900
0046          27 KSIGN(3) = -1                                                  14000
0047          28 IF ( EXSIGN ) GO TO 21                                         14100
0048            EXSIGN = .TRUE.                                                 14200
0049            GO TO 30                                                        14300
0050          29 IF ( EXPFLG ) GO TO 21                                         14400
0051            IF ( .NOT. DIGFLG ) GO TO 21                                    14500
0052          30 EXPFLG = .TRUE.                                                14600
0053            K = 3                                                           14700
0054            NPLASV = NPLART                                                 14800
0055          31 CONTINUE                                                       14900
                C*                                                              15000
                C       THE NUMBER HAS BEEN SEPARATED INTO INTEGER, FRACTION, AND 15100
                C       EXPONENT PARTS.  COMBINE THEM TO FORM THE NUMBER IN FLOATING 15200
                C       POINT.                                                  15300
                C                                                               15400
0056            IF ( EXPFLG ) GO TO 32                                          15500
0057            EXPON = 1.                                                      15600
0058            GO TO 35                                                        15700
                C*                                                              15800
                C       CALCULATE EXPONENT.  AN EXPONENT MAY BE ONLY TWO DIGITS LONG 15900
                C       AND LESS THAN 38 IN MAGNITUDE.                          16000
                C*                                                              16100
0059          32 IF ( NPLART-NPLASV-4 ) 33,33,21                                16200
0060          33 IEXPON = INTEGR(3)*KSIGN(3)                                    16300
0061            IF ( IABS( IEXPON ) - 37 ) 34,34,21                             16400
0062          34 EXPON = 10.**IEXPON                                            16500
0063            NPLART = NPLASV                                                 16600
                C*                                                              16700
                C*                                  CALCULATE MANTISSA          16800
                C*                                                              16900
0064          35 RTSHFT = 10.**NPLART                                           17000
0065            RESULT(1) = FLOAT( INTEGR(1)*KSIGN(1) )                         17100
```



```
0066            RESULT(2) = FLOAT( INTEGR(2)*KSIGN(1) ) / RTSHFT                17200
0067            ANS = (RESULT(1) + RESULT(2) )*EXPON                            17300
0068          41 RETURN                                                         17400
                C*                                                              17500
                C       ILLEGAL CHARACTER OR BAD SYNTAX.                        17600
                C*                                                              17700
0069          21 N = -1                                                         17800
0070            GO TO 41                                                        17900
0071            END                                                             18000
```

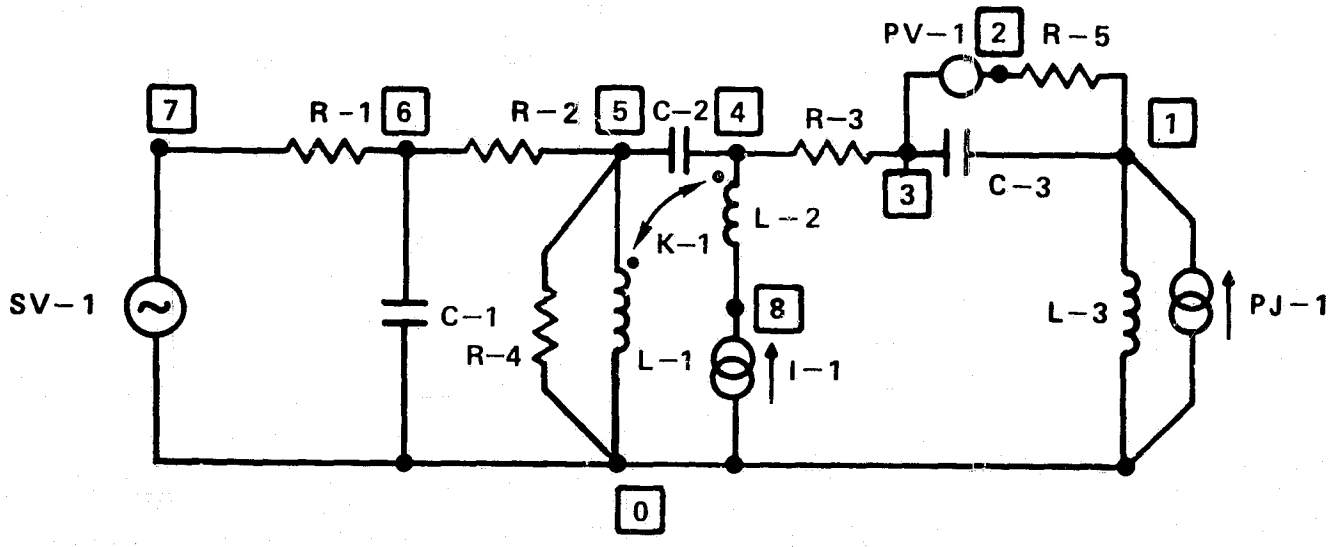
```
0001          SUBROUTINE LINK2                                                  298000
              C           CALLED BY CIRCUS                                      298100
              C       SUBROUTINE LINK2 CONTROLS READING, INTERPRETING, AND      298200
              C       STORING THE INPUT DATA.                                   298300
0002          LOGICAL ENDFLG, HOLD, HOLD2, MODLIN                               298400
0003          INTEGER TRLIST, DILIST, FETLST, TDLIST, TITLE, PEAKGM             298500
0004          INTEGER DIDICT, FETDCT, TDDICT, TRDICT                            298600
0005          DIMENSION WORD(1),NWORD(1),MNSAVE(12)                             298700
0006          DIMENSION NJFT(1), LOCBB(1), NT1(1)                               298800
0007          COMMON  WORD                                                      298900
0008          COMMON  N1    , N5    , N6    , LPDS  , LDS   , DUMMY(12)         299000
0009          COMMON  NTOT  , NJV   , NJSV  , NJPV  , NJCL  , NJC               299100
0010          COMMON  NJR   , NJL   , NJPJ  , NJM   , NJT   , NJTL              299200
0011          COMMON  NJD   , NJDL  , LINTVL, LGMDOT, LPLTVL, PEAKGM            299300
0012          COMMON  ENDFLG, N2    , N3    , NEL   , NODES , NBRNCH            299400
0013          COMMON  NCHORD, NJB   , NYSIZE, NXSIZE, NZSIZE, LELNAM            299500
0014          COMMON  LELVAL, LAMAT , NJTHET, NJDELT, NJEPS , NJZETA            299600
0015          COMMON  NJALPH, NJBETA, NJGAM , NJPHI , LENA  , LEVA              299700
0016          COMMON  LENB  , LEVB  , LENG  , LEVG  , LENP  , LEVP              299800
0017          COMMON  LENT  , LEVT  , LEND  , LEVD  , LENE  , LEVE              299900
0018          COMMON  LENZ  , LEVZ  , LTRIP , LMTPAR, LTRPAR, LTDPAR            300000
0019          COMMON  LDPAR , LSVOLT, LPVOLT, LPCURR, LCLEAK, LTRLST            300100
0020          COMMON  LDILST, LZGG  , LZGD  , LZDG  , LZDD  , LSDGP             300200
0021          COMMON  LSDDT , LSGP  , LSGT  , LSDP  , LSDT  , LQGG              300300
0022          COMMON  LPDD  , LSCR  , LFPDD , LFQGG , LFQBB , LFPEE             300400
0023          COMMON  NVSAVE, LVHOLD, LVALIC, LVALIP, LFSDGP, LFSDDT            300500
0024          COMMON  LFSGP , LFSGT , NPCELL, LPLT  , LLZZ  , LCZGD             300600
0025          COMMON  LCZPD , LCEG  , LCED  , LCEP  , LCET  , LFCZGD            300700
0026          COMMON  N4    , LFCEG , LFCED , LFCEP , LFCET , N7                300800
0027          COMMON  NJMT  , NJMTL , NJTD  , NJTDL , INDDEV, MINT              300900
0028          COMMON  LSPC  , TMAX  , TNEXT , NRGIP , NTIP  , LCC               301000
0029          COMMON  LRATNG, LMAXMS, NMAXMS, NLC   , NTPV  , JXX               301100
0030          COMMON  NTPJ  , JIPP  , LCHNGE, KSPCAL, NVSRCE, NSWV              301200
0031          COMMON  NPV   , NVDS  , NVE   , NVZ   , NJSRCE, NDSWV             301300
0032          COMMON  NDPV  , NDJSRC, NSL   , NSRCE , NSX   , NSFT              301400
0033          COMMON  NVL   , NVD   , NVX   , LFLAGX, NVWORK, NDVC              301500
0034          COMMON  LPBB  , LTEMP , NPPRNT, NPPLOT, NALZ  , NNV               301600
0035          COMMON  NPRNT , NIZ   , NIBCD , NPAGE , NPNAME, NCUR              301700
0036          COMMON  NELP  , NSPR  , NELG  , NSG   , NJPULS, LLINK4            301800
0037          COMMON  NOIPP , LF    , LR    , LRSTAB, NMODLS , NJX              301900
0038          COMMON  LSCPM , NXC   , NEQ   , NGDL  , NGD   , LTRSET            302000
0039          COMMON  INDMDT, IPLOTI, NPLOTS, KPRNT , NPNT  , DTCUR             302100
0040          COMMON  LAX   , LEN   , LPHI  , LTHA  , LXIS  , LTC1              302200
0041          COMMON  LMDL  , LLOCB , LLOCT , LNBN  , LNTC  , LXI               302300
0042          COMMON  XLISTX(14784), NELTYP(500), JNEL(500)                     302400
0043          EQUIVALENCE (WORD(1),NWORD(1),NZERO)                              302500
0044          COMMON /EXTCOM/  NJFT,  NJZD,  NJFTL,  NJZDL, LFTPAR, LZDPAR     302600
             1      , LFTLST, LZDLST                                           302700
0045          COMMON/DICTCM/FETDCT(21),TRDICT(38),DIDICT(20),TDDICT(19),        302800
             1             MTDICT(47),ZDDICT                                  302900
0046          COMMON /MDLNAM/ TRLIST(25), DILIST(25), MTRLST(5), TDLIST(25)     303000
             1             , FETLST(25), ZDLIST(25)                           303100
0047          COMMON /MISC  / KFT,HOLD,LREST,NPEAK,TIME,IRATE,LSAVE,L,NBIAS     303200
             1     , JNODES(6), JPRINT(6), JPLOT(6), PLTINT, HOLD2, JFUNCT     303300
```

```
0048            COMMON /SCRTCH/ SKIP(99),KOUNT,ZNAME(100)                       303400
0049            COMMON /MTXEQ1/ LOCBB, LOCTCB, THETMT(10), XISMT(10), NBB(10),  303500
               1                NTCB(10)                                        303600
0050            COMMON /VINDEX/    NT1,    NT2,    ND1,    ND2,    NZD1,   NZD2 303700
               1       ,   NFT1,   NFT2,   NMT1,   NMT2,   NTD1,   NTD2        303800
0051            COMMON /COMGIC/ NGUESS, NGTRUE, GUESIC(70)                      303900
0052            COMMON /TITLE / TITLE(20)                                       304000
                C                                                               344800
                C   * EXECUTE STATEMENT.                                       344900
                C   * SEE IF ANY ILLEGAL STATEMENTS PRECEDE THE EXECUTE STATEMENT 345000
                C   * IN 'RESTART' MODE; GO DIRECTLY TO TRANSIENT ANALYSIS.     345100
0053            WRITE(6,1)
0054          1 FORMAT(' ENTERED LINK2')
0055         36 IF ( .NOT. ENDFLG ) GO TO 3500                                  345200
0056            IF ( KLOCK(X) .LE. 0 ) GO TO 35
0057            IF ( NOIPP  )'37,39,37                                          345400
0058         37 IF ( PEAKGM ) 39,38,39                                          345500
0059         38 WRITE (N6,1005)                                                 345600
0060            GO TO 35                                                        345700
0061         39 CONTINUE                                                        345800
0062            IF ( HOLD2 ) GO TO 551                                          345900
0063            IRATE = 1                                                       346000
0064            NJPULS = NJPV+NJPJ                                              346100
0065            IF (LCHNGE.EQ.2) GO TO 210                                      346200
0066            LCHNGE = 2                                                      346300
0067            NMODLS = NJTL+NJTL+NJDL+NJZDL+NJFTL+NJFTL+3*NJMTL+NJTDL         346400
              C*****************************************************************346500
                C                                                               346600
                C   * INPUT DEVICE PARAMETERS FROM TAPE.                       346700
                C   * (SKIP THIS SECTION IF NO DEVICES IN CIRCUIT).            346800
0068            IF ( NMODLS ) 405,406,405                                       346900
0069        405 IF ( MODLIN ) GO TO 406                                         347000
0070            GO TO 3501
                C                                                               344100
                C   * UNDEFINED STATEMENT.                                     344200
0071        380 WRITE (N6,1004)                                                 344300
0072         35 ENDFLG = .FALSE.                                                344400
0073            WRITE(6,1010)
0074            RETURN
0075       3500 WRITE (N6,1006)                                                 344600
0076            WRITE(6,1010)
0077            RETURN
                C   * STORE STARTING ADDRESS OF TUNNEL DIODE PARAMETER ARRAY.  347100
0078       3501 LTDPAR=LPDS
0079            LPDS=LPDS+10*NJTD                                               347300
0080            CALL EQVLNT                                                     347400
0081            IF ( .NOT. ENDFLG ) GO TO 35                                    347500
0082            MODLIN = .TRUE.                                                 347600
                C       IF ( LCHNGE ) 406,376,406          CHANGE WHEN GET 'CHANGE'TO GO
                C   * CONSTRUCT 'PACKED' TOPOLOGICAL MATRICES, ALSO            347800
                C     ELEMENT NAME AND VALUE LISTS.                             347900
0083        406 CALL INCIDN                                                     348000
0084            IF ( .NOT. ENDFLG ) GO TO 35                                    348100
                C   * ALLOCATE PERMANENT DYNAMIC STORAGE.                       348200
```

```
0085            CALL MAPPER                                                     348300
0086            IF ( .NOT. ENDFLG ) GO TO 35                                    348400
              C*************************************************************** 348500
              C                                                                 348600
              C   *     TRANSFER ALL REQUIRED DATA INTO PERMANENT STORAGE.       348700
              C         MOVE THE DEVICE PARALLEL MULTIPLIERS.                   348800
              C****                                                             348900
              C*************************************************************** 349000
              C****                                                             349100
              C**** SUBROUTINE LINK2 HAS GOTTEN TOO BIG TO COMPILE ON THE IBM-7094 349200
              C****  COMPUTER.  THEREFORE, THE CODE WHICH WAS PREVIOUSLY ON CARDS 349300
              C**** BETWEEN SEQUENCE NUMBERS CIR24510 AND CIR25060 HAS NOW BEEN 349400
              C**** TRANSFERRED TO SUBROUTINE MAPPER.                           349500
              C****                                                             349600
              C*************************************************************** 349700
              C****                                                             349800
0087            210 CONTINUE                                                    349900
0088            IF (NJPULS.EQ.0) GO TO 551                                      350000
              C   * MOVE PULSED SOURCE START, RISE, DURATION, AND FALL          350100
              C     TIMES INTO WORKING AREA.                                    350200
0089            DO 550 I=1, NJPULS                                              350300
0090            LIST = LPVOLT+15*I                                              350400
0091            NWORD(LIST-2) = 1                                               350500
0092            550 CALL EQUAT1 (4, WORD(LIST-8), WORD(LIST-13))                350600
0093            551 CONTINUE                                                    350700
0094            JFUNCT = 1                                                      350800
0095            RETURN                                                          350900
0096          1004 FORMAT(29H0**  UNDEFINED STATEMENT.  **//)                   351300
0097          1005 FORMAT(47H0**  PHOTOCURRENTS WERE INPUT BUT NO PEAK RATE.//) 351400
0098          1006 FORMAT(76H0**  EXECUTION SUPPRESSED DUE TO PREVIOUS ERRORS.  ERROR
                  1 SCAN CONTINUES.  **//)                                      351500
                                                                                351600
0099          1010 FORMAT(' ***   ERROR EARLY IN TRANSIENT ANALYSIS (LINK2)   ***')
0100            END                                                             352000
```

APPENDIX B

SAMPLE PROBLEM NO. 1



NODE	(7,6)	R-1 = 110
NODE	(6,5)	R-2 = 120
NODE	(4,3)	R-3 = 130
NODE	(5,0)	R-4 = 140
NODE	(2,1)	R-5 = 150
NODE	(6,0)	C-1 = 2.1 E-6
NODE	(5,4)	C-2 = 2.2 E-6
NODE	(3,1)	C-3 = 2.3 E-7
NODE	(5,0)	L-1 = 0.031
NODE	(8,4)	L-2 = 0.032
NODE	(1,0)	L-3 = 0.033
NODE	(1,2)	K-1 = -0.8
NODE	(7,0)	SV-1 = 0.04, 0.04, 0.003, 0.004
NODE	(3,2)	PV-1 = 0.005, 0.005, 0, 0, 10, 0, 10
NODE	(1,0)	PJ-1 = 0, 0.06, 0, 0, 0.0005, 0, 5
NODE	(8,0)	I-1 = 0.007
Time	0,0	0.100 E-4, 0,100 E-2

TIME (USEC)	IR1	IR2	IR3	IR4	IR5	IC1	IC2	IC3	IL1	IL2	IL3
0.0	3.48E-04	3.48E-04	-7.00E-03	0.0	-7.00E-03	-7.16E-19	8.75E-19	-1.18E-18	3.48E-04	-7.00E-03	-7.00E-03
10.000	1.65E-03	2.96E-02	4.95E-02	-2.61E-02	-5.34E-03	-2.79E-02	5.65E-02	5.48E-02	-8.66E-04	-7.00E-03	-1.05E-02
20.000	2.77E-03	2.66E-02	4.60E-02	-2.44E-02	-3.82E-03	-2.38E-02	5.30E-02	4.98E-02	-2.01E-03	-7.00E-03	-1.40E-02
30.000	3.72E-03	2.38E-02	4.26E-02	-2.28E-02	-2.44E-03	-2.01E-02	4.96E-02	4.51E-02	-3.07E-03	-7.00E-03	-1.74E-02
40.000	4.51E-03	2.12E-02	3.93E-02	-2.11E-02	-1.20E-03	-1.66E-02	4.63E-02	4.05E-02	-4.06E-03	-7.00E-03	-2.07E-02
50.000	5.16E-03	1.86E-02	3.61E-02	-1.95E-02	-9.31E-05	-1.35E-02	4.31E-02	3.62E-02	-4.98E-03	-7.00E-03	-2.39E-02
60.000	5.68E-03	1.63E-02	3.29E-02	-1.78E-02	8.95E-04	-1.06E-02	3.99E-02	3.20E-02	-5.82E-03	-7.00E-03	-2.71E-02
70.000	6.08E-03	1.40E-02	2.99E-02	-1.63E-02	1.77E-03	-7.97E-03	3.69E-02	2.81E-02	-6.59E-03	-7.00E-03	-3.01E-02
80.000	6.37E-03	1.20E-02	2.69E-02	-1.47E-02	2.53E-03	-5.58E-03	3.39E-02	2.44E-02	-7.29E-03	-7.00E-03	-3.31E-02
90.000	6.57E-03	1.00E-02	2.41E-02	-1.32E-02	3.18E-03	-3.43E-03	3.11E-02	2.09E-02	-7.92E-03	-7.00E-03	-3.59E-02
100.000	6.67E-03	8.17E-03	2.13E-02	-1.17E-02	3.74E-03	-1.50E-03	2.83E-02	1.76E-02	-8.48E-03	-7.00E-03	-3.87E-02
110.000	6.70E-03	6.47E-03	1.87E-02	-1.02E-02	4.21E-03	2.30E-04	2.57E-02	1.45E-02	-8.97E-03	-7.00E-03	-4.13E-02
120.000	6.65E-03	4.89E-03	1.61E-02	-8.85E-03	4.58E-03	1.76E-03	2.31E-02	1.16E-02	-9.40E-03	-7.00E-03	-4.39E-02
130.000	6.55E-03	3.43E-03	1.37E-02	-7.52E-03	4.88E-03	3.12E-03	2.07E-02	8.84E-03	-9.77E-03	-7.00E-03	-4.63E-02
140.000	6.38E-03	2.08E-03	1.14E-02	-6.23E-03	5.10E-03	4.30E-03	1.84E-02	6.30E-03	-1.01E-02	-7.00E-03	-4.86E-02
150.000	6.17E-03	8.41E-04	9.19E-03	-5.01E-03	5.24E-03	5.33E-03	1.62E-02	3.94E-03	-1.03E-02	-7.00E-03	-5.08E-02
160.000	5.92E-03	-2.93E-04	7.08E-03	-3.84E-03	5.33E-03	6.21E-03	1.41E-02	1.76E-03	-1.05E-02	-7.00E-03	-5.29E-02
170.000	5.63E-03	-1.33E-03	5.09E-03	-2.73E-03	5.35E-03	6.96E-03	1.21E-02	-2.60E-04	-1.07E-02	-7.00E-03	-5.49E-02
180.000	5.32E-03	-2.26E-03	3.20E-03	-1.68E-03	5.31E-03	7.58E-03	1.02E-02	-2.11E-03	-1.08E-02	-7.00E-03	-5.68E-02
190.000	4.98E-03	-3.11E-03	1.41E-03	-6.86E-04	5.23E-03	8.09E-03	8.41E-03	-3.81E-03	-1.08E-02	-7.00E-03	-5.86E-02
200.000	4.62E-03	-3.87E-03	-2.69E-04	2.47E-04	5.09E-03	8.49E-03	6.73E-03	-5.36E-03	-1.08E-02	-7.00E-03	-6.03E-02
210.000	4.24E-03	-4.55E-03	-1.85E-03	1.12E-03	4.92E-03	8.79E-03	5.15E-03	-6.77E-03	-1.08E-02	-7.00E-03	-6.19E-02
220.000	3.85E-03	-5.14E-03	-3.34E-03	1.94E-03	4.70E-03	9.00E-03	3.66E-03	-8.04E-03	-1.07E-02	-7.00E-03	-6.33E-02
230.000	3.46E-03	-5.67E-03	-4.72E-03	2.70E-03	4.45E-03	9.13E-03	2.28E-03	-9.18E-03	-1.06E-02	-7.00E-03	-6.47E-02
240.000	3.06E-03	-6.13E-03	-6.02E-03	3.40E-03	4.17E-03	9.19E-03	9.80E-04	-1.02E-02	-1.05E-02	-7.00E-03	-6.60E-02
250.000	2.66E-03	-6.52E-03	-7.23E-03	4.05E-03	3.86E-03	9.18E-03	-2.26E-04	-1.11E-02	-1.03E-02	-7.00E-03	-6.72E-02
259.999	2.26E-03	-6.85E-03	-8.35E-03	4.64E-03	3.53E-03	9.11E-03	-1.35E-03	-1.19E-02	-1.01E-02	-7.00E-03	-6.83E-02
269.999	1.87E-03	-7.12E-03	-9.38E-03	5.18E-03	3.18E-03	8.99E-03	-2.38E-03	-1.26E-02	-9.92E-03	-7.00E-03	-6.94E-02
279.999	1.48E-03	-7.34E-03	-1.03E-02	5.67E-03	2.80E-03	8.82E-03	-3.33E-03	-1.31E-02	-9.67E-03	-7.00E-03	-7.03E-02
289.999	1.10E-03	-7.51E-03	-1.12E-02	6.11E-03	2.41E-03	8.61E-03	-4.21E-03	-1.36E-02	-9.41E-03	-7.00E-03	-7.12E-02
299.999	7.30E-04	-7.63E-03	-1.20E-02	6.51E-03	2.01E-03	8.36E-03	-5.01E-03	-1.40E-02	-9.12E-03	-7.00E-03	-7.20E-02
309.999	3.72E-04	-7.71E-03	-1.27E-02	6.85E-03	1.60E-03	8.08E-03	-5.74E-03	-1.43E-02	-8.82E-03	-7.00E-03	-7.27E-02
319.998	2.56E-05	-7.75E-03	-1.34E-02	7.16E-03	1.18E-03	7.78E-03	-6.41E-03	-1.46E-02	-8.50E-03	-7.00E-03	-7.34E-02
329.998	-3.07E-04	-7.75E-03	-1.40E-02	7.42E-03	7.57E-04	7.45E-03	-7.00E-03	-1.48E-02	-8.17E-03	-7.00E-03	-7.40E-02
339.998	-6.25E-04	-7.73E-03	-1.45E-02	7.65E-03	3.28E-04	7.10E-03	-7.54E-03	-1.49E-02	-7.83E-03	-7.00E-03	-7.45E-02
349.998	-9.27E-04	-7.67E-03	-1.50E-02	7.83E-03	-1.04E-04	6.74E-03	-8.02E-03	-1.49E-02	-7.48E-03	-7.00E-03	-7.50E-02
359.998	-1.21E-03	-7.58E-03	-1.54E-02	7.98E-03	-5.36E-04	6.37E-03	-8.44E-03	-1.49E-02	-7.13E-03	-7.00E-03	-7.54E-02
369.998	-1.48E-03	-7.47E-03	-1.58E-02	8.10E-03	-9.67E-04	5.99E-03	-8.81E-03	-1.48E-02	-6.76E-03	-7.00E-03	-7.58E-02
379.998	-1.74E-03	-7.34E-03	-1.61E-02	8.18E-03	-1.40E-03	5.60E-03	-9.13E-03	-1.47E-02	-6.40E-03	-7.00E-03	-7.61E-02
389.997	-1.98E-03	-7.19E-03	-1.64E-02	8.24E-03	-1.82E-03	5.21E-03	-9.40E-03	-1.46E-02	-6.02E-03	-7.00E-03	-7.64E-02
399.997	-2.20E-03	-7.02E-03	-1.66E-02	8.26E-03	-2.24E-03	4.82E-03	-9.63E-03	-1.44E-02	-5.65E-03	-7.00E-03	-7.66E-02
409.997	-2.40E-03	-6.84E-03	-1.68E-02	8.26E-03	-2.65E-03	4.44E-03	-9.82E-03	-1.42E-02	-5.28E-03	-7.00E-03	-7.68E-02
419.997	-2.59E-03	-6.64E-03	-1.70E-02	8.24E-03	-3.06E-03	4.05E-03	-9.97E-03	-1.39E-02	-4.91E-03	-7.00E-03	-7.70E-02
429.997	-2.76E-03	-6.43E-03	-1.71E-02	8.19E-03	-3.46E-03	3.67E-03	-1.01E-02	-1.36E-02	-4.54E-03	-7.00E-03	-7.71E-02
439.997	-2.91E-03	-6.21E-03	-1.72E-02	8.12E-03	-3.85E-03	3.30E-03	-1.02E-02	-1.33E-02	-4.17E-03	-7.00E-03	-7.72E-02
449.996	-3.05E-03	-5.99E-03	-1.72E-02	8.03E-03	-4.23E-03	2.93E-03	-1.02E-02	-1.30E-02	-3.80E-03	-7.00E-03	-7.72E-02
459.996	-3.17E-03	-5.75E-03	-1.72E-02	7.93E-03	-4.60E-03	2.58E-03	-1.02E-02	-1.26E-02	-3.44E-03	-7.00E-03	-7.72E-02
469.996	-3.28E-03	-5.52E-03	-1.72E-02	7.80E-03	-4.96E-03	2.23E-03	-1.02E-02	-1.23E-02	-3.09E-03	-7.00E-03	-7.72E-02
479.996	-3.38E-03	-5.28E-03	-1.72E-02	7.67E-03	-5.31E-03	1.90E-03	-1.02E-02	-1.19E-02	-2.74E-03	-7.00E-03	-7.72E-02
489.996	-3.46E-03	-5.03E-03	-1.72E-02	7.52E-03	-5.65E-03	1.58E-03	-1.02E-02	-1.15E-02	-2.39E-03	-7.00E-03	-7.72E-02

TIME (USEC)	PJ1	PJ2	VR1	VR2	VR3	VR4	VR5	VC1	VC2	VC3	VL1
0.0	0.0	7.00E-03	3.83E-02	4.17E-02	-0.910	0.0	-1.050	4.17E-02	1.955	-1.045	0.0
10.000	6.00E-02	7.00E-03	0.181	3.547	6.433	-3.649	-0.800	-0.102	2.220	-0.795	-3.649
20.000	6.00E-02	7.00E-03	0.305	3.193	5.984	-3.418	-0.573	-0.225	2.469	-0.568	-3.418
30.000	6.00E-02	7.00E-03	0.409	2.857	5.543	-3.186	-0.367	-0.329	2.702	-0.362	-3.186
40.000	6.00E-02	7.00E-03	0.496	2.539	5.112	-2.955	-0.181	-0.416	2.920	-0.176	-2.955
50.000	6.00E-02	7.00E-03	0.568	2.237	4.692	-2.725	-1.40E-02	-0.488	3.123	-8.96E-03	-2.725
60.000	6.00E-02	7.00E-03	0.625	1.953	4.283	-2.498	0.134	-0.545	3.312	0.139	-2.498
70.000	6.00E-02	7.00E-03	0.669	1.686	3.886	-2.275	0.265	-0.589	3.486	0.270	-2.275
80.000	6.00E-02	7.00E-03	0.701	1.435	3.501	-2.056	0.379	-0.622	3.647	0.384	-2.056
90.000	6.00E-02	7.00E-03	0.722	1.200	3.130	-1.843	0.478	-0.643	3.795	0.483	-1.843
100.000	6.00E-02	7.00E-03	0.734	0.981	2.773	-1.635	0.561	-0.655	3.930	0.566	-1.635
110.000	6.00E-02	7.00E-03	0.737	0.776	2.429	-1.434	0.631	-0.657	4.053	0.636	-1.434
120.000	6.00E-02	7.00E-03	0.732	0.587	2.099	-1.239	0.687	-0.653	4.164	0.692	-1.239
130.000	6.00E-02	7.00E-03	0.720	0.411	1.783	-1.052	0.732	-0.641	4.263	0.737	-1.052
140.000	6.00E-02	7.00E-03	0.702	0.250	1.482	-0.873	0.764	-0.623	4.352	0.769	-0.873
150.000	6.00E-02	7.00E-03	0.679	0.101	1.194	-0.701	0.787	-0.600	4.431	0.792	-0.701
160.000	6.00E-02	7.00E-03	0.651	-3.51E-02	0.921	-0.538	0.799	-0.573	4.500	0.804	-0.538
170.000	6.00E-02	7.00E-03	0.620	-0.159	0.661	-0.382	0.802	-0.541	4.559	0.807	-0.382
180.000	6.00E-02	7.00E-03	0.585	-0.272	0.416	-0.235	0.797	-0.507	4.610	0.802	-0.235
190.000	6.00E-02	7.00E-03	0.547	-0.373	0.184	-9.60E-02	0.784	-0.469	4.652	0.789	-9.60E-02
200.000	6.00E-02	7.00E-03	0.508	-0.464	-3.50E-02	3.46E-02	0.764	-0.430	4.686	0.769	3.46E-02
210.000	6.00E-02	7.00E-03	0.466	-0.546	-0.241	0.157	0.738	-0.389	4.713	0.743	0.157
220.000	6.00E-02	7.00E-03	0.424	-0.617	-0.434	0.271	0.705	-0.346	4.733	0.710	0.271
230.000	6.00E-02	7.00E-03	0.380	-0.680	-0.614	0.377	0.668	-0.303	4.747	0.673	0.377
240.000	6.00E-02	7.00E-03	0.337	-0.735	-0.783	0.476	0.626	-0.259	4.754	0.631	0.476
250.000	6.00E-02	7.00E-03	0.293	-0.782	-0.939	0.566	0.579	-0.216	4.756	0.584	0.566
259.999	6.00E-02	7.00E-03	0.249	-0.821	-1.085	0.649	0.529	-0.172	4.752	0.534	0.649
269.999	6.00E-02	7.00E-03	0.205	-0.854	-1.219	0.725	0.476	-0.129	4.744	0.481	0.725
279.999	6.00E-02	7.00E-03	0.163	-0.880	-1.343	0.794	0.420	-8.66E-02	4.731	0.425	0.794
289.999	6.00E-02	7.00E-03	0.121	-0.901	-1.457	0.856	0.362	-4.51E-02	4.713	0.367	0.856
299.999	6.00E-02	7.00E-03	8.03E 02	-0.915	-1.562	0.911	0.302	-4.67E-03	4.692	0.307	0.911
309.999	6.00E-02	7.00E-03	4.09E-02	-0.925	-1.657	0.960	0.240	3.45E-02	4.668	0.245	0.960
319.998	6.00E-02	7.00E-03	2.81E-03	-0.930	-1.743	1.002	0.177	7.22E-02	4.640	0.182	1.002
329.998	6.00E-02	7.00E-03	-3.37E-02	-0.931	-1.820	1.039	0.114	0.108	4.610	0.119	1.039
339.998	6.00E-02	7.00E-03	-6.87E-02	-0.927	-1.890	1.070	4.92E-02	0.143	4.577	5.42E-02	1.070
349.998	6.00E-02	7.00E-03	-0.102	-0.920	-1.952	1.096	-1.56E-02	0.176	4.541	-1.06E-02	1.096
359.998	6.00E-02	7.00E-03	-0.134	-0.910	-2.007	1.117	-8.04E-02	0.207	4.504	-7.54E-02	1.117
369.998	6.00E-02	7.00E-03	-0.163	-0.897	-2.055	1.134	-0.145	0.237	4.465	-0.140	1.134
379.998	6.00E-02	7.00E-03	-0.191	-0.881	-2.097	1.145	-0.209	0.264	4.424	-0.204	1.145
389.997	6.00E-02	7.00E-03	-0.217	-0.863	-2.132	1.153	-0.273	0.290	4.382	-0.268	1.153
399.997	6.00E-02	7.00E-03	-0.242	-0.843	-2.162	1.157	-0.336	0.314	4.338	-0.331	1.157
409.997	6.00E-02	7.00E-03	-0.264	-0.820	-2.187	1.157	-0.398	0.336	4.294	-0.393	1.157
419.997	6.00E-02	7.00E-03	-0.285	-0.797	-2.206	1.153	-0.459	0.356	4.249	-0.454	1.153
429.997	6.00E-02	7.00E-03	-0.303	-0.772	-2.221	1.146	-0.519	0.375	4.204	-0.514	1.146
439.997	6.00E-02	7.00E-03	-0.320	-0.746	-2.232	1.137	-0.578	0.391	4.158	-0.573	1.137
449.996	6.00E-02	7.00E-03	-0.336	-0.718	-2.238	1.124	-0.635	0.406	4.111	-0.630	1.124
459.996	6.00E-02	7.00E-03	-0.349	-0.690	-2.241	1.110	-0.691	0.419	4.065	-0.686	1.110
469.996	6.00E-02	7.00E-03	-0.361	-0.662	-2.241	1.093	-0.745	0.431	4.018	-0.740	1.093
479.996	6.00E-02	7.00E-03	-0.371	-0.633	-2.237	1.074	-0.797	0.441	3.972	-0.792	1.074
489.996	6.00E-02	7.00E-03	-0.380	-0.604	-2.230	1.053	-0.848	0.449	3.926	-0.843	1.053

TIME (USEC)	VL2	VL3	SV1	PV1	VPJ1	VPJ2
0.0	9.54E-07	5.96E-08	8.00E-02	5.00E-03	5.96E-08	-1.955
10.000	2.966	-11.51	8.00E-02	5.00E-03	-11.51	-2.903
20.000	2.778	-11.30	8.00E-02	5.00E-03	-11.30	-3.108
30.000	2.590	-11.07	8.00E-02	5.00E-03	-11.07	-3.298
40.000	2.402	-10.81	7.99E-02	5.00E-03	-10.81	-3.473
50.000	2.215	-10.53	7.99E-02	5.00E-03	-10.53	-3.633
60.000	2.031	-10.23	7.98E-02	5.00E-03	-10.23	-3.780
70.000	1.849	-9.917	7.98E-02	5.00E-03	-9.917	-3.912
80.000	1.671	-9.589	7.97E-02	5.00E-03	-9.589	-4.032
90.000	1.498	-9.251	7.96E-02	5.00E-03	-9.251	-4.140
100.000	1.329	-8.904	7.95E-02	5.00E-03	-8.904	-4.236
110.000	1.165	-8.551	7.94E-02	5.00E-03	-8.551	-4.321
120.000	1.007	-8.194	7.93E-02	5.00E-03	-8.194	-4.396
130.000	0.855	-7.835	7.92E-02	5.00E-03	-7.835	-4.460
140.000	0.709	-7.476	7.90E-02	5.00E-03	-7.476	-4.516
150.000	0.570	-7.118	7.89E-02	5.00E-03	-7.118	-4.562
160.000	0.437	-6.762	7.87E-02	5.00E-03	-6.762	-4.600
170.000	0.311	-6.410	7.86E-02	5.00E-03	-6.410	-4.630
180.000	0.191	-6.062	7.84E-02	5.00E-03	-6.062	-4.654
190.000	7.80E-02	-5.721	7.82E-02	5.00E-03	-5.721	-4.670
200.000	-2.81E-02	-5.386	7.80E-02	5.00E-03	-5.386	-4.680
210.000	-0.128	-5.058	7.78E-02	5.00E-03	-5.058	-4.684
220.000	-0.220	-4.739	7.76E-02	5.00E-03	-4.739	-4.682
230.000	-0.307	-4.428	7.74E-02	5.00E-03	-4.428	-4.676
240.000	-0.387	-4.126	7.72E-02	5.00E-03	-4.126	-4.665
250.000	-0.460	-3.834	7.70E-02	5.00E-03	-3.834	-4.650
259.999	-0.528	-3.552	7.67E-02	5.00E-03	-3.552	-4.630
269.999	-0.589	-3.280	7.65E-02	5.00E-03	-3.280	-4.608
279.999	-0.645	-3.019	7.62E-02	5.00E-03	-3.019	-4.582
289.999	-0.695	-2.768	7.59E-02	5.00E-03	-2.768	-4.553
299.999	-0.740	-2.527	7.56E-02	5.00E-03	-2.527	-4.522
309.999	-0.780	-2.297	7.54E-02	5.00E-03	-2.297	-4.488
319.998	-0.815	-2.078	7.51E-02	5.00E-03	-2.078	-4.453
329.998	-0.845	-1.869	7.47E-02	5.00E-03	-1.869	-4.415
339.998	-0.870	-1.670	7.44E-02	5.00E-03	-1.670	-4.376
349.998	-0.891	-1.482	7.41E-02	5.00E-03	-1.482	-4.336
359.998	-0.908	-1.304	7.38E-02	5.00E-03	-1.304	-4.295
369.998	-0.921	-1.136	7.34E-02	5.00E-03	-1.136	-4.252
379.998	-0.931	-0.977	7.31E-02	5.00E-03	-0.977	-4.209
389.997	-0.937	-0.828	7.27E-02	5.00E-03	-0.828	-4.166
399.997	-0.940	-0.689	7.24E-02	5.00E-03	-0.689	-4.122
409.997	-0.940	-0.558	7.20E-02	5.00E-03	-0.558	-4.078
419.997	-0.937	-0.436	7.16E-02	5.00E-03	-0.436	-4.033
429.997	-0.932	-0.322	7.12E-02	5.00E-03	-0.322	-3.989
439.997	-0.924	-0.217	7.08E-02	5.00E-03	-0.217	-3.945
449.996	-0.914	-0.119	7.04E-02	5.00E-03	-0.119	-3.901
459.996	-0.902	-2.85E-02	7.00E-02	5.00E-03	-2.85E-02	-3.857
469.996	-0.888	5.46E-02	6.96E-02	5.00E-03	5.46E-02	-3.814
479.996	-0.873	0.131	6.92E-02	5.00E-03	0.131	-3.771
489.996	-0.856	0.200	6.87E-02	5.00E-03	0.200	-3.728

30 MAY 70

TIME (USEC)	IR1	IR2	IR3	IR4	IR5	IC1	IC2	IC3	IL1	IL2	IL3
499.996	-3.52E-03	-4.79E-03	-1.71E-02	7.36E-03	-5.98E-03	1.27E-03	-1.01E-02	-1.11E-02	-2.06E-03	-7.00E-03	-7.71E-02
509.996	-4.87E-03	-3.38E-02	-7.35E-02	3.32E-02	-7.96E-03	2.89E-02	-6.65E-02	-6.55E-02	-5.16E-04	-7.00E-03	-7.35E-02
519.995	-6.03E-03	-3.06E-02	-6.99E-02	3.14E-02	-9.78E-03	2.45E-02	-6.29E-02	-6.01E-02	9.44E-04	-7.00E-03	-6.99E-02
529.995	-7.01E-03	-2.75E-02	-6.64E-02	2.96E-02	-1.14E-02	2.05E-02	-5.94E-02	-5.50E-02	2.32E-03	-7.00E-03	-6.64E-02
539.995	-7.82E-03	-2.46E-02	-6.30E-02	2.77E-02	-1.30E-02	1.68E-02	-5.60E-02	-5.00E-02	3.62E-03	-7.00E-03	-6.30E-02
549.995	-8.48E-03	-2.19E-02	-5.96E-02	2.59E-02	-1.43E-02	1.34E-02	-5.26E-02	-4.52E-02	4.83E-03	-7.00E-03	-5.96E-02
559.995	-8.99E-03	-1.93E-02	-5.63E-02	2.41E-02	-1.56E-02	1.03E-02	-4.93E-02	-4.07E-02	5.95E-03	-7.00E-03	-5.63E-02
569.995	-9.38E-03	-1.68E-02	-5.31E-02	2.22E-02	-1.67E-02	7.43E-03	-4.61E-02	-3.64E-02	7.00E-03	-7.00E-03	-5.31E-02
579.994	-9.65E-03	-1.45E-02	-4.99E-02	2.05E-02	-1.77E-02	4.85E-03	-4.29E-02	-3.22E-02	7.96E-03	-7.00E-03	-4.99E-02
589.994	-9.81E-03	-1.23E-02	-4.69E-02	1.87E-02	-1.86E-02	2.51E-03	-3.99E-02	-2.83E-02	8.85E-03	-7.00E-03	-4.69E-02
599.994	-9.88E-03	-1.03E-02	-4.40E-02	1.70E-02	-1.93E-02	4.00E-04	-3.70E-02	-2.46E-02	9.66E-03	-7.00E-03	-4.40E-02
609.994	-9.86E-03	-8.37E-03	-4.11E-02	1.54E-02	-2.00E-02	-1.49E-03	-3.41E-02	-2.11E-02	1.04E-02	-7.00E-03	-4.11E-02
619.994	-9.76E-03	-6.59E-03	-3.84E-02	1.38E-02	-2.06E-02	-3.17E-03	-3.14E-02	-1.78E-02	1.10E-02	-7.00E-03	-3.84E-02
629.994	-9.60E-03	-4.93E-03	-3.58E-02	1.22E-02	-2.10E-02	-4.67E-03	-2.88E-02	-1.47E-02	1.16E-02	-7.00E-03	-3.58E-02
639.993	-9.37E-03	-3.40E-03	-3.33E-02	1.07E-02	-2.14E-02	-5.98E-03	-2.63E-02	-1.18E-02	1.21E-02	-7.00E-03	-3.33E-02
649.993	-9.09E-03	-1.98E-03	-3.08E-02	9.27E-03	-2.17E-02	-7.12E-03	-2.38E-02	-9.12E-03	1.26E-02	-7.00E-03	-3.08E-02
659.993	-8.77E-03	-6.68E-04	-2.85E-02	7.89E-03	-2.20E-02	-8.10E-03	-2.15E-02	-6.59E-03	1.30E-02	-7.00E-03	-2.85E-02
669.993	-8.40E-03	5.33E-04	-2.64E-02	6.57E-03	-2.21E-02	-8.94E-03	-1.94E-02	-4.24E-03	1.33E-02	-7.00E-03	-2.64E-02
679.993	-8.01E-03	1.63E-03	-2.43E-02	5.32E-03	-2.22E-02	-9.64E-03	-1.73E-02	-2.06E-03	1.36E-02	-7.00E-03	-2.43E-02
689.993	-7.58E-03	2.63E-03	-2.23E-02	4.12E-03	-2.22E-02	-1.02E-02	-1.53E-02	-4.61E-05	1.38E-02	-7.00E-03	-2.23E-02
699.993	-7.13E-03	3.54E-03	-2.04E-02	2.99E-03	-2.22E-02	-1.07E-02	-1.34E-02	1.81E-03	1.40E-02	-7.00E-03	-2.04E-02
709.992	-6.67E-03	4.35E-03	-1.86E-02	1.92E-03	-2.21E-02	-1.10E-02	-1.16E-02	3.50E-03	1.41E-02	-7.00E-03	-1.86E-02
719.992	-6.19E-03	5.09E-03	-1.70E-02	9.12E-04	-2.20E-02	-1.13E-02	-9.96E-03	5.05E-03	1.41E-02	-7.00E-03	-1.70E-02
729.992	-5.70E-03	5.74E-03	-1.54E-02	-3.27E-05	-2.18E-02	-1.14E-02	-8.38E-03	6.46E-03	1.41E-02	-7.00E-03	-1.54E-02
739.992	-5.21E-03	6.31E-03	-1.39E-02	-9.16E-04	-2.16E-02	-1.15E-02	-6.90E-03	7.74E-03	1.41E-02	-7.00E-03	-1.39E-02
749.992	-4.72E-03	6.82E-03	-1.25E-02	-1.74E-03	-2.14E-02	-1.15E-02	-5.51E-03	8.88E-03	1.41E-02	-7.00E-03	-1.25E-02
759.992	-4.22E-03	7.25E-03	-1.12E-02	-2.51E-03	-2.11E-02	-1.15E-02	-4.21E-03	9.91E-03	1.40E-02	-7.00E-03	-1.12E-02
769.991	-3.74E-03	7.62E-03	-1.00E-02	-3.21E-03	-2.08E-02	-1.14E-02	-3.01E-03	1.08E-02	1.38E-02	-7.00E-03	-1.00E-02
779.991	-3.25E-03	7.94E-03	-8.88E-03	-3.86E-03	-2.05E-02	-1.12E-02	-1.88E-03	1.16E-02	1.37E-02	-7.00E-03	-8.88E-03
789.991	-2.78E-03	8.19E-03	-7.84E-03	-4.46E-03	-2.01E-02	-1.10E-02	-8.40E-04	1.23E-02	1.35E-02	-7.00E-03	-7.84E-03
799.991	-2.31E-03	8.40E-03	-6.88E-03	-5.01E-03	-1.98E-02	-1.07E-02	1.24E-04	1.29E-02	1.33E-02	-7.00E-03	-6.88E-03
809.991	-1.86E-03	8.55E-03	-5.99E-03	-5.50E-03	-1.94E-02	-1.04E-02	1.01E-03	1.34E-02	1.30E-02	-7.00E-03	-5.99E-03
819.991	-1.42E-03	8.67E-03	-5.17E-03	-5.94E-03	-1.90E-02	-1.01E-02	1.83E-03	1.38E-02	1.28E-02	-7.00E-03	-5.17E-03
829.990	-1.00E-03	8.74E-03	-4.43E-03	-6.34E-03	-1.86E-02	-9.74E-03	2.57E-03	1.42E-02	1.25E-02	-7.00E-03	-4.43E-03
839.990	-5.92E-04	8.77E-03	-3.75E-03	-6.69E-03	-1.82E-02	-9.36E-03	3.25E-03	1.44E-02	1.22E-02	-7.00E-03	-3.75E-03
849.990	-2.01E-04	8.77E-03	-3.13E-03	-7.00E-03	-1.78E-02	-8.97E-03	3.87E-03	1.46E-02	1.19E-02	-7.00E-03	-3.13E-03
859.990	1.73E-04	8.73E-03	-2.57E-03	-7.27E-03	-1.73E-02	-8.56E-03	4.43E-03	1.48E-02	1.16E-02	-7.00E-03	-2.57E-03
869.990	5.29E-04	8.67E-03	-2.07E-03	-7.50E-03	-1.69E-02	-8.14E-03	4.93E-03	1.48E-02	1.12E-02	-7.00E-03	-2.07E-03
879.990	8.66E-04	8.58E-03	-1.63E-03	-7.69E-03	-1.65E-02	-7.71E-03	5.37E-03	1.49E-02	1.09E-02	-7.00E-03	-1.63E-03
889.990	1.19E-03	8.47E-03	-1.23E-03	-7.85E-03	-1.60E-02	-7.28E-03	5.77E-03	1.48E-02	1.06E-02	-7.00E-03	-1.23E-03
899.989	1.49E-03	8.33E-03	-8.86E-04	-7.98E-03	-1.56E-02	-6.84E-03	6.11E-03	1.47E-02	1.02E-02	-7.00E-03	-8.86E-04
909.989	1.77E-03	8.18E-03	-5.85E-04	-8.07E-03	-1.52E-02	-6.41E-03	6.42E-03	1.46E-02	9.83E-03	-7.00E-03	-5.85E-04
919.989	2.03E-03	8.01E-03	-3.25E-04	-8.13E-03	-1.48E-02	-5.98E-03	6.67E-03	1.44E-02	9.47E-03	-7.00E-03	-3.25E-04
929.989	2.27E-03	7.82E-03	-1.06E-04	-8.17E-03	-1.44E-02	-5.55E-03	6.89E-03	1.43E-02	9.10E-03	-7.00E-03	-1.06E-04
939.989	2.50E-03	7.62E-03	7.67E-05	-8.18E-03	-1.39E-02	-5.12E-03	7.08E-03	1.40E-02	8.73E-03	-7.00E-03	7.67E-05
949.989	2.71E-03	7.41E-03	2.24E-04	-8.17E-03	-1.35E-02	-4.71E-03	7.22E-03	1.38E-02	8.36E-03	-7.00E-03	2.24E-04
959.988	2.89E-03	7.19E-03	3.40E-04	-8.14E-03	-1.31E-02	-4.30E-03	7.34E-03	1.35E-02	7.99E-03	-7.00E-03	3.40E-04
969.988	3.07E-03	6.97E-03	4.25E-04	-8.08E-03	-1.28E-02	-3.90E-03	7.43E-03	1.32E-02	7.62E-03	-7.00E-03	4.25E-04
979.988	3.22E-03	6.73E-03	4.83E-04	-8.01E-03	-1.24E-02	-3.51E-03	7.48E-03	1.29E-02	7.26E-03	-7.00E-03	4.83E-04
989.988	3.36E-03	6.50E-03	5.15E-04	-7.92E-03	-1.20E-02	-3.14E-03	7.52E-03	1.25E-02	6.90E-03	-7.00E-03	5.15E-04

TIME (USEC)	PJ1	PJ2	VR1	VR2	VR3	VR4	VR5	VC1	VC2	VC3	VL1
499.996	6.00E-02	7.00E-03	-0.387	-0.574	-2.221	1.030	-0.897	0.456	3.880	-0.892	1.030
509.996	0.0	7.00E-03	-0.536	-4.050	-9.553	4.655	-1.194	0.604	3.569	-1.189	4.655
519.995	0.0	7.00E-03	-0.664	-3.667	-9.090	4.398	-1.467	0.731	3.275	-1.462	4.398
529.995	0.0	7.00E-03	-0.771	-3.302	-8.634	4.140	-1.717	0.838	2.997	-1.712	4.140
539.995	0.0	7.00E-03	-0.860	-2.955	-8.185	3.882	-1.945	0.927	2.735	-1.940	3.882
549.995	0.0	7.00E-03	-0.933	-2.625	-7.746	3.624	-2.152	0.999	2.488	-2.147	3.624
559.995	0.0	7.00E-03	-0.989	-2.313	-7.317	3.368	-2.339	1.055	2.257	-2.334	3.368
569.995	0.0	7.00E-03	-1.032	-2.018	-6.898	3.115	-2.506	1.097	2.040	-2.501	3.115
579.994	0.0	7.00E-03	-1.062	-1.740	-6.492	2.866	-2.655	1.126	1.838	-2.650	2.866
589.994	0.0	7.00E-03	-1.080	-1.479	-6.097	2.622	-2.787	1.144	1.650	-2.782	2.622
599.994	0.0	7.00E-03	-1.087	-1.234	-5.716	2.384	-2.902	1.150	1.475	-2.897	2.384
609.994	0.0	7.00E-03	-1.085	-1.005	-5.347	2.152	-3.001	1.148	1.314	-2.996	2.152
619.994	0.0	7.00E-03	-1.074	-0.791	-4.992	1.927	-3.086	1.137	1.165	-3.081	1.927
629.994	0.0	7.00E-03	-1.056	-0.592	-4.651	1.710	-3.157	1.118	1.028	-3.152	1.710
639.993	0.0	7.00E-03	-1.031	-0.408	-4.324	1.500	-3.214	1.092	0.903	-3.209	1.500
649.993	0.0	7.00E-03	-1.000	-0.237	-4.010	1.298	-3.260	1.061	0.789	-3.255	1.298
659.993	0.0	7.00E-03	-0.964	-8.01E-02	-3.711	1.105	-3.294	1.025	0.686	-3.289	1.105
669.993	0.0	7.00E-03	-0.924	6.40E-02	-3.426	0.920	-3.317	0.984	0.593	-3.312	0.920
679.993	0.0	7.00E-03	-0.881	0.196	-3.154	0.744	-3.331	0.940	0.510	-3.326	0.744
689.993	0.0	7.00E-03	-0.834	0.316	-2.897	0.577	-3.336	0.893	0.436	-3.331	0.577
699.993	0.0	7.00E-03	-0.785	0.425	-2.653	0.418	-3.332	0.843	0.371	-3.327	0.418
709.992	0.0	7.00E-03	-0.734	0.523	-2.422	0.269	-3.320	0.791	0.314	-3.315	0.269
719.992	0.0	7.00E-03	-0.681	0.610	-2.204	0.128	-3.301	0.738	0.265	-3.296	0.128
729.992	0.0	7.00E-03	-0.627	0.689	-1.999	-4.58E-03	-3.276	0.684	0.223	-3.271	-4.58E-03
739.992	0.0	7.00E-03	-0.573	0.758	-1.807	-0.128	-3.245	0.629	0.188	-3.240	-0.128
749.992	0.0	7.00E-03	-0.519	0.818	-1.626	-0.244	-3.209	0.574	0.160	-3.204	-0.244
759.992	0.0	7.00E-03	-0.465	0.870	-1.458	-0.351	-3.168	0.519	0.138	-3.163	-0.351
769.991	0.0	7.00E-03	-0.411	0.915	-1.301	-0.450	-3.123	0.465	0.122	-3.118	-0.450
779.991	0.0	7.00E-03	-0.358	0.952	-1.155	-0.541	-3.074	0.411	0.111	-3.069	-0.541
789.991	0.0	7.00E-03	-0.306	0.983	-1.019	-0.625	-3.022	0.359	0.105	-3.017	-0.625
799.991	0.0	7.00E-03	-0.255	1.008	-0.894	-0.701	-2.968	0.307	0.103	-2.963	-0.701
809.991	0.0	7.00E-03	-0.205	1.027	-0.778	-0.770	-2.910	0.257	0.106	-2.905	-0.770
819.991	0.0	7.00E-03	-0.157	1.040	-0.673	-0.832	-2.851	0.208	0.112	-2.846	-0.832
829.990	0.0	7.00E-03	-0.110	1.048	-0.576	-0.888	-2.790	0.161	0.122	-2.785	-0.888
839.990	0.0	7.00E-03	-6.52E-02	1.052	-0.487	-0.937	-2.728	0.115	0.135	-2.723	-0.937
849.990	0.0	7.00E-03	-2.21E-02	1.052	-0.407	-0.981	-2.665	7.15E-02	0.152	-2.660	-0.981
859.990	0.0	7.00E-03	1.90E-02	1.048	-0.335	-1.018	-2.601	2.97E-02	0.171	-2.596	-1.018
869.990	0.0	7.00E-03	5.82E-02	1.040	-0.270	-1.050	-2.536	-1.01E-02	0.192	-2.531	-1.050
879.990	0.0	7.00E-03	9.53E-02	1.029	-0.212	-1.077	-2.472	-4.78E-02	0.215	-2.467	-1.077
889.990	0.0	7.00E-03	0.130	1.016	-0.160	-1.099	-2.407	-8.35E-02	0.241	-2.402	-1.099
899.989	0.0	7.00E-03	0.163	1.000	-0.115	-1.117	-2.343	-0.117	0.268	-2.338	-1.117
909.989	0.0	7.00E-03	0.194	0.981	-7.60E-02	-1.130	-2.279	-0.149	0.296	-2.274	-1.130
919.989	0.0	7.00E-03	0.223	0.961	-4.23E-02	-1.139	-2.216	-0.178	0.326	-2.211	-1.139
929.989	0.0	7.00E-03	0.250	0.938	-1.37E-02	-1.144	-2.154	-0.206	0.357	-2.149	-1.144
939.989	0.0	7.00E-03	0.275	0.915	9.97E-03	-1.146	-2.092	-0.231	0.388	-2.087	-1.146
949.989	0.0	7.00E-03	0.298	0.889	2.92E-02	-1.144	-2.032	-0.254	0.421	-2.027	-1.144
959.988	0.0	7.00E-03	0.318	0.863	4.42E-02	-1.139	-1.972	-0.276	0.454	-1.967	-1.139
969.988	0.0	7.00E-03	0.337	0.836	5.53E-02	-1.131	-1.914	-0.295	0.488	-1.909	-1.131
979.988	0.0	7.00E-03	0.354	0.808	6.28E-02	-1.121	-1.858	-0.313	0.522	-1.853	-1.121
989.988	0.0	7.00E-03	0.369	0.780	6.70E-02	-1.109	-1.803	-0.329	0.556	-1.798	-1.109

TIME (USEC)	VL2	VL3	SV1	PV1	VPJ1	VPJ2
499.996	-0.837	0.264	6.83E-02	5.00E-03	0.264	-3.687
509.996	-3.783	11.83	6.78E-02	5.00E-03	11.83	-2.698
519.995	-3.575	11.68	6.74E-02	5.00E-03	11.68	-2.452
529.995	-3.365	11.49	6.69E-02	5.00E-03	11.49	-2.222
539.995	-3.155	11.27	6.65E-02	5.00E-03	11.27	-2.008
549.995	-2.945	11.03	6.60E-02	5.00E-03	11.03	-1.810
559.995	-2.737	10.76	6.55E-02	5.00E-03	10.76	-1.626
569.995	-2.532	10.47	6.50E-02	5.00E-03	10.47	-1.457
579.994	-2.330	10.17	6.45E-02	5.00E-03	10.17	-1.301
589.994	-2.131	9.852	6.40E-02	5.00E-03	9.852	-1.159
599.994	-1.938	9.522	6.35E-02	5.00E-03	9.522	-1.029
609.994	-1.749	9.182	6.30E-02	5.00E-03	9.182	-0.911
619.994	-1.567	8.836	6.25E-02	5.00E-03	8.836	-0.804
629.994	-1.390	8.485	6.20E-02	5.00E-03	8.485	-0.708
639.993	-1.219	8.130	6.14E-02	5.00E-03	8.130	-0.622
649.993	-1.055	7.775	6.09E-02	5.00E-03	7.775	-0.546
659.993	-0.898	7.419	6.04E-02	5.00E-03	7.419	-0.479
669.993	-0.748	7.066	5.98E-02	5.00E-03	7.066	-0.421
679.993	-0.605	6.715	5.93E-02	5.00E-03	6.715	-0.370
689.993	-0.469	6.368	5.87E-02	5.00E-03	6.368	-0.328
699.993	-0.340	6.027	5.82E-02	5.00E-03	6.027	-0.292
709.992	-0.218	5.692	5.76E-02	5.00E-03	5.692	-0.264
719.992	-0.104	5.363	5.70E-02	5.00E-03	5.363	-0.241
729.992	3.72E-03	5.043	5.65E-02	5.00E-03	5.043	-0.224
739.992	0.104	4.730	5.59E-02	5.00E-03	4.730	-0.213
749.992	0.198	4.427	5.53E-02	5.00E-03	4.427	-0.206
759.992	0.285	4.132	5.47E-02	5.00E-03	4.132	-0.204
769.991	0.366	3.847	5.41E-02	5.00E-03	3.847	-0.206
779.991	0.440	3.572	5.36E-02	5.00E-03	3.572	-0.212
789.991	0.508	3.307	5.30E-02	5.00E-03	3.307	-0.222
799.991	0.570	3.053	5.24E-02	5.00E-03	3.053	-0.234
809.991	0.626	2.808	5.18E-02	5.00E-03	2.808	-0.250
819.991	0.676	2.574	5.12E-02	5.00E-03	2.574	-0.268
829.990	0.722	2.351	5.06E-02	5.00E-03	2.351	-0.288
839.990	0.762	2.137	4.99E-02	5.00E-03	2.137	-0.311
849.990	0.797	1.934	4.93E-02	5.00E-03	1.934	-0.335
859.990	0.828	1.742	4.87E-02	5.00E-03	1.742	-0.361
869.990	0.854	1.559	4.81E-02	5.00E-03	1.559	-0.388
879.990	0.876	1.386	4.75E-02	5.00E-03	1.386	-0.417
889.990	0.894	1.223	4.69E-02	5.00E-03	1.223	-0.446
899.989	0.908	1.069	4.63E-02	5.00E-03	1.069	-0.477
909.989	0.918	0.924	4.56E-02	5.00E-03	0.924	-0.508
919.989	0.926	0.789	4.50E-02	5.00E-03	0.789	-0.539
929.989	0.930	0.662	4.44E-02	5.00E-03	0.662	-0.571
939.989	0.931	0.543	4.38E-02	5.00E-03	0.543	-0.603
949.989	0.930	0.433	4.31E-02	5.00E-03	0.433	-0.635
959.988	0.926	0.330	4.25E-02	5.00E-03	0.330	-0.667
969.988	0.920	0.235	4.19E-02	5.00E-03	0.235	-0.699
979.988	0.911	0.147	4.13E-02	5.00E-03	0.147	-0.731
989.988	0.901	6.65E-02	4.06E-02	5.00E-03	6.65E-02	-0.763
END L5	9999997					

30 MAY 70

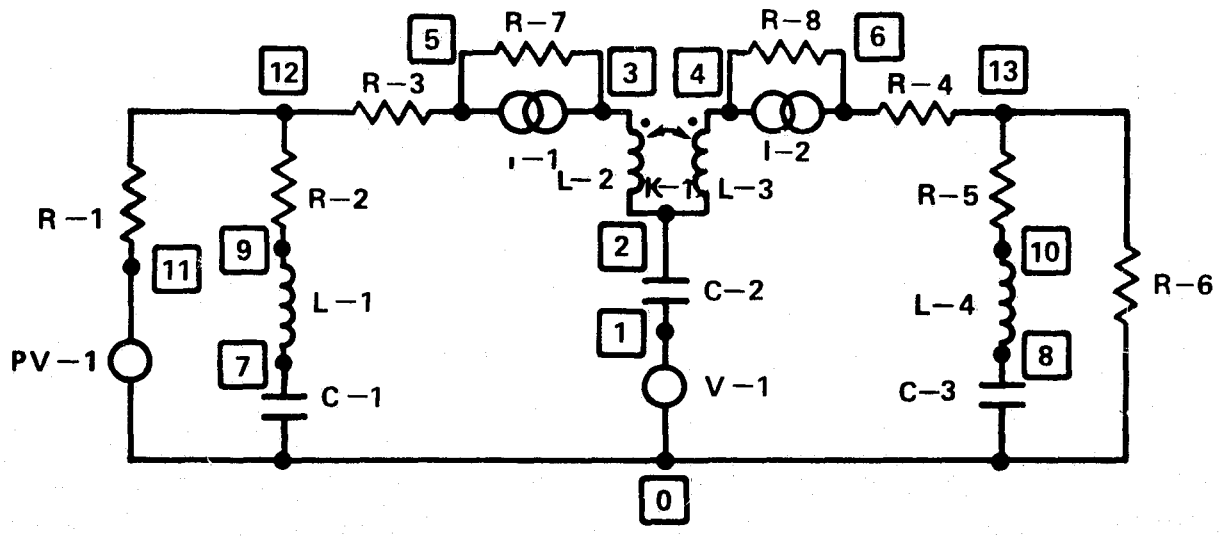
TIME (USEC)	IR1	IR2	IR3	IR4	IR5	IC1	IC2	IC3	IL1	IL2	IL3
999.988	3.48E-03	6.26E-03	5.24E-04	-7.81E-03	-1.17E-02	-2.78E-03	7.52E-03	1.22E-02	6.55E-03	-7.00E-03	5.24E-04

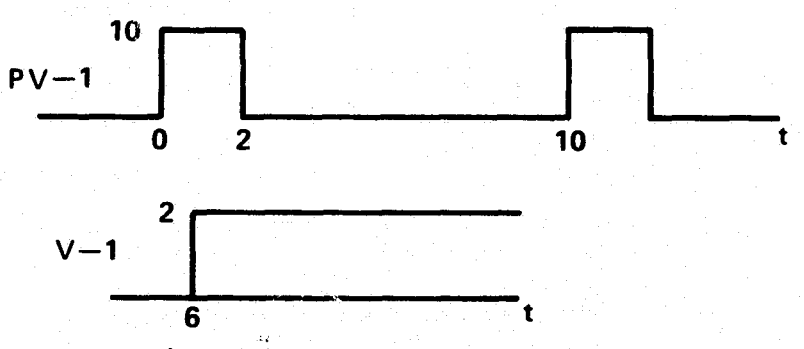
TIME (USEC)	PJ1	PJ2	VR1	VR2	VR3	VR4	VR5	VC1	VC2	VC3	VL1
999.988	0.0	7.00E-03	0.383	0.751	6.81E-02	-1.094	-1.749	-0.343	0.590	-1.744	-1.094

TIME (USEC)	VL2	VL3	SV1	PV1	VPJ1	VPJ2
999.988	0.889	-7.78E-03	4.00E-02	5.00E-03	-7.78E-03	-0.795

ENDJ08 9999997

SAMPLE PROBLEM NO. 2






```
NODE      (11,0)    PV-1 = 0, 10, 0, 0, 2E-3, 0, 1E-3
NODE       (1,0)     V-1 = 2
NODE      (12,11)   R-1 = 50
NODE      (12,9)    R-2 = 5E-4
NODE      (12,5)    R-3 = 100
NODE      (13,6)    R-4 = 10
NODE      (13,10)   R-5 = 5E-4
NODE      (13,0)    R-6 = 5E-3
NODE       (5,3)    R-7 = 10
NODE       (6,4)    R-8 = 10
NODE       (9,7)    L-1 = 1.0E-3
NODE       (3,2)    L-2 = 12E-3
NODE       (4,2)    L-3 = 12E-3
NODE      (10,8)    L-4 = 1.0E-3
NODE       (7,0)    C-1 = 0.05E-6
NODE       (8,0)    C-3 = 0.05E-6
NODE       (5,3)     I-1 = 1.0E-3
NODE       (6,4)     I-2 = 1.5E-3
NODE       (2,3)    K-1 = 0.95
```

TIME (USEC)	IR1	IR2	IR3	IR4	IR5	IR6	IR7	IR8	IL1	IL2	IL3
0.0	9.49E-07	2.91E-16	-9.49E-07	9.49E-07	-7.45E-14	-9.49E-07	-1.00E-03	-1.50E-03	3.53E-26	-9.49E-07	9.49E-07
10.000	-9.42E-03	1.90E-04	9.23E-03	-1.74E-03	1.58E-04	1.58E-03	8.23E-03	-3.24E-03	1.90E-04	9.23E-03	-1.74E-03
20.000	-1.58E-02	1.83E-04	1.57E-02	-1.53E-03	1.38E-04	1.39E-03	1.47E-02	-3.03E-03	1.83E-04	1.57E-02	-1.53E-03
30.000	-2.14E-02	1.76E-04	2.12E-02	-1.35E-03	1.21E-04	1.23E-03	2.02E-02	-2.85E-03	1.76E-04	2.12E-02	-1.35E-03
40.000	-2.62E-02	1.71E-04	2.61E-02	-1.19E-03	1.06E-04	1.08E-03	2.51E-02	-2.69E-03	1.71E-04	2.61E-02	-1.19E-03
50.000	-3.04E-02	1.66E-04	3.02E-02	-1.06E-03	9.37E-05	9.62E-04	2.92E-02	-2.56E-03	1.66E-04	3.02E-02	-1.06E-03
60.000	-3.39E-02	1.62E-04	3.38E-02	-9.41E-04	8.30E-05	8.58E-04	3.28E-02	-2.44E-03	1.62E-04	3.38E-02	-9.41E-04
70.000	-3.69E-02	1.58E-04	3.68E-02	-8.45E-04	7.39E-05	7.71E-04	3.58E-02	-2.34E-03	1.58E-04	3.68E-02	-8.45E-04
80.000	-3.94E-02	1.55E-04	3.93E-02	-7.65E-04	6.64E-05	6.98E-04	3.83E-02	-2.26E-03	1.55E-04	3.93E-02	-7.65E-04
90.000	-4.14E-02	1.52E-04	4.13E-02	-7.00E-04	6.02E-05	6.39E-04	4.03E-02	-2.20E-03	1.52E-04	4.13E-02	-7.00E-04
100.000	-4.31E-02	1.50E-04	4.29E-02	-6.48E-04	5.53E-05	5.92E-04	4.19E-02	-2.15E-03	1.50E-04	4.29E-02	-6.48E-04
110.000	-4.44E-02	1.48E-04	4.42E-02	-6.08E-04	5.15E-05	5.56E-04	4.32E-02	-2.11E-03	1.48E-04	4.42E-02	-6.08E-04
120.000	-4.53E-02	1.47E-04	4.52E-02	-5.78E-04	4.86E-05	5.30E-04	4.42E-02	-2.08E-03	1.47E-04	4.52E-02	-5.78E-04
130.000	-4.60E-02	1.45E-04	4.59E-02	-5.58E-04	4.66E-05	5.12E-04	4.49E-02	-2.06E-03	1.45E-04	4.59E-02	-5.58E-04
140.000	-4.64E-02	1.44E-04	4.63E-02	-5.47E-04	4.54E-05	5.01E-04	4.53E-02	-2.05E-03	1.44E-04	4.63E-02	-5.47E-04
150.000	-4.66E-02	1.44E-04	4.65E-02	-5.42E-04	4.48E-05	4.97E-04	4.55E-02	-2.04E-03	1.44E-04	4.65E-02	-5.42E-04
160.000	-4.66E-02	1.43E-04	4.65E-02	-5.44E-04	4.48E-05	4.99E-04	4.55E-02	-2.04E-03	1.43E-04	4.65E-02	-5.44E-04
170.000	-4.64E-02	1.43E-04	4.63E-02	-5.52E-04	4.54E-05	5.07E-04	4.53E-02	-2.05E-03	1.43E-04	4.63E-02	-5.52E-04
180.000	-4.61E-02	1.42E-04	4.60E-02	-5.65E-04	4.64E-05	5.19E-04	4.50E-02	-2.06E-03	1.42E-04	4.60E-02	-5.65E-04
190.000	-4.57E-02	1.42E-04	4.55E-02	-5.82E-04	4.78E-05	5.34E-04	4.45E-02	-2.08E-03	1.42E-04	4.55E-02	-5.82E-04
200.000	-4.51E-02	1.42E-04	4.49E-02	-6.03E-04	4.95E-05	5.54E-04	4.39E-02	-2.10E-03	1.42E-04	4.49E-02	-6.03E-04
210.000	-3.50E-02	-4.73E-05	3.50E-02	1.12E-03	-1.06E-04	-1.01E-03	3.40E-02	-3.84E-04	-4.73E-05	3.50E-02	1.12E-03
220.000	-2.78E-02	-4.00E-05	2.79E-02	8.78E-04	-8.45E-05	-7.94E-04	2.69E-02	-6.22E-04	-4.00E-05	2.79E-02	8.78E-04
230.000	-2.14E-02	-3.34E-05	2.14E-02	6.67E-04	-6.49E-05	-6.02E-04	2.04E-02	-8.33E-04	-3.34E-05	2.14E-02	6.67E-04
240.000	-1.57E-02	-2.76E-05	1.57E-02	4.78E-04	-4.76E-05	-4.31E-04	1.47E-02	-1.02E-03	-2.76E-05	1.57E-02	4.78E-04
250.000	-1.06E-02	-2.24E-05	1.06E-02	3.11E-04	-3.22E-05	-2.78E-04	9.63E-03	-1.19E-03	-2.24E-05	1.06E-02	3.11E-04
259.999	-6.09E-03	-1.78E-05	6.11E-03	1.62E-04	-1.86E-05	-1.43E-04	5.11E-03	-1.34E-03	-1.78E-05	6.11E-03	1.62E-04
269.999	-2.10E-03	-1.37E-05	2.12E-03	3.11E-05	-6.64E-06	-2.44E-05	1.12E-03	-1.47E-03	-1.37E-05	2.12E-03	3.11E-05
279.999	1.41E-03	-1.02E-05	-1.40E-03	-8.42E-05	3.84E-06	8.03E-05	-2.40E-03	-1.58E-03	-1.02E-05	-1.40E-03	-8.42E-05
289.999	4.49E-03	-7.07E-06	-4.49E-03	-1.85E-04	1.30E-05	1.72E-04	-5.49E-03	-1.68E-03	-7.07E-06	-4.49E-03	-1.85E-04
299.999	7.18E-03	-4.35E-06	-7.18E-03	-2.73E-04	2.09E-05	2.52E-04	-8.18E-03	-1.77E-03	-4.35E-06	-7.18E-03	-2.73E-04
309.999	9.52E-03	-2.01E-06	-9.52E-03	-3.49E-04	2.77E-05	3.21E-04	-1.05E-02	-1.85E-03	-2.01E-06	-9.52E-03	-3.49E-04
319.998	1.15E-02	1.39E-08	-1.15E-02	-4.14E-04	3.36E-05	3.81E-04	-1.25E-02	-1.91E-03	1.39E-08	-1.15E-02	-4.14E-04
329.998	1.33E-02	1.74E-06	-1.33E-02	-4.70E-04	3.85E-05	4.32E-04	-1.43E-02	-1.97E-03	1.74E-06	-1.33E-02	-4.70E-04
339.998	1.47E-02	3.19E-06	-1.47E-02	-5.17E-04	4.26E-05	4.74E-04	-1.57E-02	-2.02E-03	3.19E-06	-1.47E-02	-5.17E-04
349.998	1.60E-02	4.40E-06	-1.60E-02	-5.56E-04	4.60E-05	5.10E-04	-1.70E-02	-2.06E-03	4.40E-06	-1.60E-02	-5.56E-04
359.998	1.70E-02	5.40E-06	-1.70E-02	-5.89E-04	4.88E-05	5.40E-04	-1.80E-02	-2.09E-03	5.40E-06	-1.70E-02	-5.89E-04
369.998	1.78E-02	6.21E-06	-1.78E-02	-6.15E-04	5.10E-05	5.64E-04	-1.88E-02	-2.11E-03	6.21E-06	-1.78E-02	-6.15E-04
379.998	1.85E-02	6.84E-06	-1.85E-02	-6.35E-04	5.27E-05	5.82E-04	-1.95E-02	-2.14E-03	6.84E-06	-1.85E-02	-6.35E-04
389.997	1.90E-02	7.31E-06	-1.90E-02	-6.50E-04	5.39E-05	5.97E-04	-2.00E-02	-2.15E-03	7.31E-06	-1.90E-02	-6.50E-04
399.997	1.93E-02	7.65E-06	-1.93E-02	-6.61E-04	5.47E-05	6.07E-04	-2.03E-02	-2.16E-03	7.65E-06	-1.93E-02	-6.61E-04
409.997	1.96E-02	7.87E-06	-1.96E-02	-6.68E-04	5.51E-05	6.13E-04	-2.06E-02	-2.17E-03	7.87E-06	-1.96E-02	-6.68E-04
419.997	1.97E-02	7.98E-06	-1.97E-02	-6.72E-04	5.52E-05	6.16E-04	-2.07E-02	-2.17E-03	7.98E-06	-1.97E-02	-6.72E-04
429.997	1.98E-02	7.99E-06	-1.98E-02	-6.72E-04	5.50E-05	6.17E-04	-2.08E-02	-2.17E-03	7.99E-06	-1.98E-02	-6.72E-04
439.997	1.97E-02	7.93E-06	-1.97E-02	-6.69E-04	5.46E-05	6.15E-04	-2.07E-02	-2.17E-03	7.93E-06	-1.97E-02	-6.69E-04
449.996	1.96E-02	7.78E-06	-1.96E-02	-6.65E-04	5.40E-05	6.11E-04	-2.06E-02	-2.16E-03	7.78E-06	-1.96E-02	-6.65E-04
459.996	1.94E-02	7.58E-06	-1.95E-02	-6.58E-04	5.31E-05	6.05E-04	-2.05E-02	-2.16E-03	7.58E-06	-1.95E-02	-6.58E-04
469.996	1.92E-02	7.32E-06	-1.92E-02	-6.49E-04	5.22E-05	5.97E-04	-2.02E-02	-2.15E-03	7.32E-06	-1.92E-02	-6.49E-04
479.996	1.89E-02	7.02E-06	-1.89E-02	-6.39E-04	5.10E-05	5.88E-04	-1.99E-02	-2.14E-03	7.02E-06	-1.89E-02	-6.39E-04
489.996	1.86E-02	6.67E-06	-1.86E-02	-6.27E-04	4.98E-05	5.77E-04	-1.96E-02	-2.13E-03	6.67E-06	-1.86E-02	-6.27E-04

TIME (USEC)	IL4	IC1	IC2	IC3	PJ1	PJ2	PV1	VR1	VR2	VR3	VR4
0.0	-2.88E-22	3.53E-26	-1.05E-21	-2.88E-22	1.00E-03	1.50E-03	0.0	4.74E-05	1.46E-11	-9.49E-05	9.49E-05
10.000	1.58E-04	1.90E-04	7.49E-03	1.58E-04	1.00E-03	1.50E-03	10.00	-0.471	9.491	0.923	-0.174
20.000	1.38E-04	1.83E-04	1.41E-02	1.38E-04	1.00E-03	1.50E-03	10.00	-0.792	9.133	1.565	-0.153
30.000	1.21E-04	1.76E-04	1.99E-02	1.21E-04	1.00E-03	1.50E-03	10.00	-1.071	8.818	2.124	-0.135
40.000	1.06E-04	1.71E-04	2.49E-02	1.06E-04	1.00E-03	1.50E-03	10.00	-1.312	8.542	2.608	-0.119
50.000	9.37E-05	1.66E-04	2.92E-02	9.37E-05	1.00E-03	1.50E-03	10.00	-1.520	8.300	3.023	-0.106
60.000	8.30E-05	1.62E-04	3.28E-02	8.30E-05	1.00E-03	1.50E-03	10.00	-1.697	8.091	3.377	-9.41E-02
70.000	7.39E-05	1.58E-04	3.59E-02	7.39E-05	1.00E-03	1.50E-03	10.00	-1.846	7.909	3.676	-8.45E-02
80.000	6.64E-05	1.55E-04	3.85E-02	6.64E-05	1.00E-03	1.50E-03	10.00	-1.970	7.754	3.925	-7.65E-02
90.000	6.02E-05	1.52E-04	4.06E-02	6.02E-05	1.00E-03	1.50E-03	10.00	-2.072	7.621	4.129	-7.00E-02
100.000	5.53E-05	1.50E-04	4.23E-02	5.53E-05	1.00E-03	1.50E-03	10.00	-2.154	7.509	4.294	-6.48E-02
110.000	5.15E-05	1.48E-04	4.36E-02	5.15E-05	1.00E-03	1.50E-03	10.00	-2.218	7.415	4.422	-6.08E-02
120.000	4.86E-05	1.47E-04	4.46E-02	4.86E-05	1.00E-03	1.50E-03	10.00	-2.266	7.337	4.518	-5.78E-02
130.000	4.66E-05	1.45E-04	4.53E-02	4.66E-05	1.00E-03	1.50E-03	10.00	-2.300	7.274	4.585	-5.58E-02
140.000	4.54E-05	1.44E-04	4.57E-02	4.54E-05	1.00E-03	1.50E-03	10.00	-2.321	7.224	4.628	-5.47E-02
150.000	4.48E-05	1.44E-04	4.59E-02	4.48E-05	1.00E-03	1.50E-03	10.00	-2.331	7.186	4.647	-5.42E-02
160.000	4.48E-05	1.43E-04	4.59E-02	4.48E-05	1.00E-03	1.50E-03	10.00	-2.331	7.157	4.647	-5.44E-02
170.000	4.54E-05	1.43E-04	4.57E-02	4.54E-05	1.00E-03	1.50E-03	10.00	-2.322	7.137	4.630	-5.52E-02
180.000	4.64E-05	1.42E-04	4.54E-02	4.64E-05	1.00E-03	1.50E-03	10.00	-2.306	7.125	4.597	-5.65E-02
190.000	4.78E-05	1.42E-04	4.49E-02	4.78E-05	1.00E-03	1.50E-03	10.00	-2.283	7.119	4.551	-5.82E-02
200.000	4.95E-05	1.42E-04	4.43E-02	4.95E-05	1.00E-03	1.50E-03	10.00	-2.254	7.119	4.494	-6.03E-02
210.000	-1.06E-04	-4.73E-05	3.62E-02	-1.06E-04	1.00E-03	1.50E-03	0.0	-1.749	-2.367	3.504	0.112
220.000	-8.45E-05	-4.00E-05	2.87E-02	-8.45E-05	1.00E-03	1.50E-03	0.0	-1.391	-1.999	2.786	8.78E-02
230.000	-6.49E-05	-3.34E-05	2.21E-02	-6.49E-05	1.00E-03	1.50E-03	0.0	-1.070	-1.671	2.144	6.67E-02
240.000	-4.76E-05	-2.76E-05	1.62E-02	-4.76E-05	1.00E-03	1.50E-03	0.0	-0.785	-1.379	1.572	4.78E-02
250.000	-3.22E-05	-2.24E-05	1.09E-02	-3.22E-05	1.00E-03	1.50E-03	0.0	-0.530	-1.120	1.063	3.11E-02
259.999	-1.86E-05	-1.78E-05	6.27E-03	-1.86E-05	1.00E-03	1.50E-03	0.0	-0.305	-0.890	0.611	1.62E-02
269.999	-6.64E-06	-1.37E-05	2.15E-03	-6.64E-06	1.00E-03	1.50E-03	0.0	-0.105	-0.687	0.212	3.11E-03
279.999	3.84E-06	-1.02E-05	-1.49E-03	3.84E-06	1.00E-03	1.50E-03	0.0	7.06E-02	-0.509	-0.140	-8.42E-03
289.999	1.30E-05	-7.07E-06	-4.67E-03	1.30E-05	1.00E-03	1.50E-03	0.0	0.225	-0.353	-0.449	-1.85E-02
299.999	2.09E-05	-4.35E-06	-7.45E-03	2.09E-05	1.00E-03	1.50E-03	0.0	0.359	-0.218	-0.718	-2.73E-02
309.999	2.77E-05	-2.01E-06	-9.87E-03	2.77E-05	1.00E-03	1.50E-03	0.0	0.476	-0.100	-0.952	-3.49E-02
319.998	3.36E-05	1.39E-08	-1.19E-02	3.36E-05	1.00E-03	1.50E-03	0.0	0.577	6.97E-04	-1.154	-4.14E-02
329.998	3.85E-05	1.74E-06	-1.37E-02	3.85E-05	1.00E-03	1.50E-03	0.0	0.663	8.68E-02	-1.326	-4.70E-02
339.998	4.26E-05	3.19E-06	-1.52E-02	4.26E-05	1.00E-03	1.50E-03	0.0	0.736	0.160	-1.473	-5.17E-02
349.998	4.60E-05	4.40E-06	-1.65E-02	4.60E-05	1.00E-03	1.50E-03	0.0	0.798	0.220	-1.596	-5.56E-02
359.998	4.88E-05	5.40E-06	-1.76E-02	4.88E-05	1.00E-03	1.50E-03	0.0	0.848	0.270	-1.697	-5.89E-02
369.998	5.10E-05	6.21E-06	-1.84E-02	5.10E-05	1.00E-03	1.50E-03	0.0	0.890	0.310	-1.780	-6.15E-02
379.998	5.27E-05	6.84E-06	-1.91E-02	5.27E-05	1.00E-03	1.50E-03	0.0	0.923	0.342	-1.846	-6.35E-02
389.997	5.39E-05	7.31E-06	-1.96E-02	5.39E-05	1.00E-03	1.50E-03	0.0	0.948	0.366	-1.896	-6.50E-02
399.997	5.47E-05	7.65E-06	-2.00E-02	5.47E-05	1.00E-03	1.50E-03	0.0	0.966	0.383	-1.933	-6.61E-02
409.997	5.51E-05	7.87E-06	-2.03E-02	5.51E-05	1.00E-03	1.50E-03	0.0	0.979	0.393	-1.958	-6.68E-02
419.997	5.52E-05	7.98E-06	-2.04E-02	5.52E-05	1.00E-03	1.50E-03	0.0	0.986	0.399	-1.972	-6.72E-02
429.997	5.50E-05	7.99E-06	-2.04E-02	5.50E-05	1.00E-03	1.50E-03	0.0	0.988	0.400	-1.977	-6.72E-02
439.997	5.46E-05	7.93E-06	-2.04E-02	5.46E-05	1.00E-03	1.50E-03	0.0	0.986	0.396	-1.973	-6.69E-02
449.996	5.40E-05	7.78E-06	-2.03E-02	5.40E-05	1.00E-03	1.50E-03	0.0	0.981	0.389	-1.962	-6.65E-02
459.996	5.31E-05	7.58E-06	-2.01E-02	5.31E-05	1.00E-03	1.50E-03	0.0	0.972	0.379	-1.945	-6.58E-02
469.996	5.22E-05	7.32E-06	-1.99E-02	5.22E-05	1.00E-03	1.50E-03	0.0	0.961	0.366	-1.922	-6.49E-02
479.996	5.10E-05	7.02E-06	-1.96E-02	5.10E-05	1.00E-03	1.50E-03	0.0	0.947	0.351	-1.895	-6.39E-02
489.996	4.98E-05	6.67E-06	-1.93E-02	4.98E-05	1.00E-03	1.50E-03	0.0	0.931	0.334	-1.863	-6.27E-02

TIME (USEC)	VR5	VR6	VR7	VR8	VL1	VL2	VL3	VL4	VC1	VC2	VC3
0.0	-3.73E-09	-4.74E-03	-1.00E-02	-1.50E-02	-1.46E-11	-5.96E-08	-6.33E-08	3.73E-09	4.74E-05	-1.990	-4.74E-03
10.000	7.893	7.919	8.23E-02	-3.24E-02	-7.62E-04	8.495	8.096	-2.10E-03	3.87E-02	-1.971	2.81E-02
20.000	6.910	6.966	0.147	-3.03E-02	-6.71E-04	7.413	7.066	-1.83E-03	7.60E-02	-1.917	5.76E-02
30.000	6.056	6.138	0.202	-2.85E-02	-5.89E-04	6.434	6.132	-1.59E-03	0.112	-1.831	8.35E-02
40.000	5.318	5.423	0.251	-2.69E-02	-5.16E-04	5.548	5.288	-1.37E-03	0.147	-1.719	0.106
50.000	4.685	4.810	0.292	-2.56E-02	-4.49E-04	4.748	4.525	-1.17E-03	0.180	-1.583	0.126
60.000	4.148	4.290	0.328	-2.44E-02	-3.89E-04	4.026	3.837	-9.88E-04	0.213	-1.428	0.144
70.000	3.695	3.854	0.358	-2.34E-02	-3.36E-04	3.376	3.218	-8.27E-04	0.245	-1.256	0.160
80.000	3.319	3.492	0.383	-2.26E-02	-2.87E-04	2.792	2.661	-6.80E-04	0.276	-1.070	0.174
90.000	3.011	3.197	0.403	-2.20E-02	-2.43E-04	2.267	2.161	-5.52E-04	0.307	-0.872	0.186
100.000	2.765	2.962	0.419	-2.15E-02	-2.04E-04	1.797	1.713	-4.36E-04	0.337	-0.665	0.198
110.000	2.573	2.781	0.432	-2.11E-02	-1.70E-04	1.378	1.313	-3.32E-04	0.367	-0.450	0.208
120.000	2.430	2.648	0.442	-2.08E-02	-1.39E-04	1.003	0.956	-2.41E-04	0.397	-0.229	0.218
130.000	2.330	2.558	0.449	-2.06E-02	-1.11E-04	0.670	0.639	-1.60E-04	0.426	-4.13E-03	0.228
140.000	2.269	2.506	0.453	-2.05E-02	-8.74E-05	0.375	0.357	-8.77E-05	0.455	0.224	0.237
150.000	2.241	2.487	0.455	-2.04E-02	-6.57E-05	0.114	0.109	-2.46E-05	0.484	0.453	0.246
160.000	2.242	2.497	0.455	-2.04E-02	-4.73E-05	-0.115	-0.110	3.06E-05	0.512	0.683	0.255
170.000	2.270	2.534	0.453	-2.05E-02	-3.05E-05	-0.317	-0.302	7.81E-05	0.541	0.912	0.264
180.000	2.320	2.593	0.450	-2.06E-02	-1.64E-05	-0.492	-0.470	1.20E-04	0.569	1.140	0.273
190.000	2.389	2.672	0.445	-2.08E-02	-4.59E-06	-0.645	-0.615	1.56E-04	0.598	1.366	0.283
200.000	2.475	2.768	0.439	-2.10E-02	6.14E-06	-0.776	-0.740	1.87E-04	0.626	1.589	0.292
210.000	-5.318	-5.046	0.340	-3.84E-03	7.76E-04	-9.383	-8.944	2.32E-03	0.616	1.790	0.270
220.000	-4.223	-3.970	0.269	-6.22E-03	6.94E-04	-8.397	-8.004	2.07E-03	0.608	1.952	0.251
230.000	-3.247	-3.009	0.204	-8.33E-03	6.19E-04	-7.498	-7.146	1.84E-03	0.600	2.079	0.236
240.000	-2.379	-2.153	0.147	-1.02E-02	5.50E-04	-6.678	-6.365	1.63E-03	0.594	2.174	0.224
250.000	-1.610	-1.392	9.63E-02	-1.19E-02	4.87E-04	-5.931	-5.653	1.44E-03	0.589	2.242	0.216
259.999	-0.930	-0.717	5.11E-02	-1.34E-02	4.31E-04	-5.251	-5.005	1.27E-03	0.585	2.285	0.211
269.999	-0.332	-0.122	1.12E-02	-1.47E-02	3.80E-04	-4.633	-4.416	1.12E-03	0.582	2.305	0.209
279.999	0.192	0.402	-2.40E-02	-1.58E-02	3.33E-04	-4.072	-3.881	9.77E-04	0.580	2.307	0.209
289.999	0.649	0.860	-5.49E-02	-1.68E-02	2.91E-04	-3.563	-3.396	8.51E-04	0.578	2.291	0.210
299.999	1.045	1.260	-8.18E-02	-1.77E-02	2.52E-04	-3.102	-2.956	7.34E-04	0.577	2.261	0.214
309.999	1.386	1.606	-0.105	-1.85E-02	2.18E-04	-2.684	-2.558	6.30E-04	0.576	2.217	0.219
319.998	1.678	1.903	-0.125	-1.91E-02	1.86E-04	-2.307	-2.199	5.37E-04	0.576	2.163	0.225
329.998	1.925	2.158	-0.143	-1.97E-02	1.58E-04	-1.966	-1.874	4.52E-04	0.576	2.098	0.232
339.998	2.132	2.372	-0.157	-2.02E-02	1.33E-04	-1.659	-1.581	3.75E-04	0.577	2.026	0.240
349.998	2.302	2.552	-0.170	-2.06E-02	1.10E-04	-1.383	-1.318	3.07E-04	0.577	1.946	0.249
359.998	2.440	2.699	-0.180	-2.09E-02	8.97E-05	-1.135	-1.082	2.46E-04	0.578	1.861	0.259
369.998	2.550	2.818	-0.188	-2.11E-02	7.13E-05	-0.912	-0.870	1.92E-04	0.579	1.771	0.269
379.998	2.633	2.912	-0.195	-2.14E-02	5.50E-05	-0.713	-0.680	1.43E-04	0.581	1.677	0.279
389.997	2.693	2.983	-0.200	-2.15E-02	4.05E-05	-0.536	-0.510	9.88E-05	0.582	1.580	0.290
399.997	2.733	3.033	-0.203	-2.16E-02	2.76E-05	-0.377	-0.360	6.00E-05	0.584	1.480	0.301
409.997	2.754	3.066	-0.206	-2.17E-02	1.62E-05	-0.237	-0.226	2.60E-05	0.585	1.380	0.311
419.997	2.759	3.082	-0.207	-2.17E-02	6.08E-06	-0.113	-0.107	-4.29E-06	0.587	1.278	0.323
429.997	2.751	3.084	-0.208	-2.17E-02	-2.80E-06	-2.77E-03	-2.50E-03	-3.02E-05	0.588	1.176	0.334
439.997	2.730	3.074	-0.207	-2.17E-02	-1.06E-05	9.36E-02	8.94E-02	-5.42E-05	0.590	1.073	0.345
449.996	2.698	3.053	-0.206	-2.16E-02	-1.73E-05	0.178	0.170	-7.25E-05	0.592	0.972	0.355
459.996	2.657	3.023	-0.205	-2.16E-02	-2.32E-05	0.251	0.240	-9.01E-05	0.593	0.870	0.366
469.996	2.608	2.984	-0.202	-2.15E-02	-2.82E-05	0.315	0.300	-1.05E-04	0.595	0.770	0.377
479.996	2.552	2.939	-0.199	-2.14E-02	-3.25E-05	0.369	0.352	-1.17E-04	0.596	0.672	0.387
489.996	2.490	2.887	-0.196	-2.13E-02	-3.61E-05	0.416	0.397	-1.29E-04	0.597	0.574	0.397

TIME (USEC)	VPJ1	VPJ2
0.0	-1.00E-02	-1.50E-02
10.000	8.23E-02	-3.24E-02
20.000	0.147	-3.03E-02
30.000	0.202	-2.85E-02
40.000	0.251	-2.69E-02
50.000	0.292	-2.56E-02
60.000	0.328	-2.44E-02
70.000	0.358	-2.34E-02
80.000	0.383	-2.26E-02
90.000	0.403	-2.20E-02
100.000	0.419	-2.15E-02
110.000	0.432	-2.11E-02
120.000	0.442	-2.08E-02
130.000	0.449	-2.06E-02
140.000	0.453	-2.05E-02
150.000	0.455	-2.04E-02
160.000	0.455	-2.04E-02
170.000	0.453	-2.05E-02
180.000	0.450	-2.06E-02
190.000	0.445	-2.08E-02
200.000	0.439	-2.10E-02
210.000	0.340	-3.84E-03
220.000	0.269	-6.22E-03
230.000	0.204	-8.33E-03
240.000	0.147	-1.02E-02
250.000	9.63E-02	-1.19E-02
259.999	5.11E-02	-1.34E-02
269.999	1.12E-02	-1.47E-02
279.999	-2.40E-02	-1.58E-02
289.999	-5.49E-02	-1.68E-02
299.999	-8.18E-02	-1.77E-02
309.999	-0.105	-1.85E-02
319.998	-0.125	-1.91E-02
329.998	-0.143	-1.97E-02
339.998	-0.157	-2.02E-02
349.998	-0.170	-2.06E-02
359.998	-0.180	-2.09E-02
369.998	-0.188	-2.11E-02
379.998	-0.195	-2.14E-02
389.997	-0.200	-2.15E-02
399.997	-0.203	-2.16E-02
409.997	-0.206	-2.17E-02
419.997	-0.207	-2.17E-02
429.997	-0.208	-2.17E-02
439.997	-0.207	-2.17E-02
449.996	-0.206	-2.16E-02
459.996	-0.205	-2.16E-02
469.996	-0.202	-2.15E-02
479.996	-0.199	-2.14E-02
489.996	-0.196	-2.13E-02

03 JUN 70

TIME (USEC)	IR1	IR2	IR3	IR4	IR5	IR6	IR7	IR8	IL1	IL2	IL3
499.996	1.83E-02	6.30E-06	-1.83E-02	-6.15E-04	4.85E-05	5.66E-04	-1.93E-02	-2.11E-03	6.30E-06	-1.83E-02	-6.15E-04
509.996	1.79E-02	5.89E-06	-1.79E-02	-6.01E-04	4.71E-05	5.54E-04	-1.89E-02	-2.10E-03	5.89E-06	-1.79E-02	-6.01E-04
519.995	1.75E-02	5.46E-06	-1.75E-02	-5.87E-04	4.56E-05	5.41E-04	-1.85E-02	-2.09E-03	5.46E-06	-1.75E-02	-5.87E-04
529.995	1.71E-02	5.02E-06	-1.71E-02	-5.72E-04	4.41E-05	5.28E-04	-1.81E-02	-2.07E-03	5.02E-06	-1.71E-02	-5.72E-04
539.995	1.66E-02	4.56E-06	-1.66E-02	-5.56E-04	4.25E-05	5.14E-04	-1.76E-02	-2.06E-03	4.56E-06	-1.66E-02	-5.56E-04
549.995	1.62E-02	4.09E-06	-1.62E-02	-5.40E-04	4.09E-05	4.99E-04	-1.72E-02	-2.04E-03	4.09E-06	-1.62E-02	-5.40E-04
559.995	1.57E-02	3.62E-06	-1.57E-02	-5.24E-04	3.93E-05	4.85E-04	-1.67E-02	-2.02E-03	3.62E-06	-1.57E-02	-5.24E-04
569.995	1.52E-02	3.14E-06	-1.52E-02	-5.08E-04	3.77E-05	4.70E-04	-1.62E-02	-2.01E-03	3.14E-06	-1.52E-02	-5.08E-04
579.994	1.48E-02	2.66E-06	-1.48E-02	-4.92E-04	3.61E-05	4.56E-04	-1.58E-02	-1.99E-03	2.66E-06	-1.48E-02	-4.92E-04
589.994	1.43E-02	2.18E-06	-1.43E-02	-4.75E-04	3.45E-05	4.41E-04	-1.53E-02	-1.98E-03	2.18E-06	-1.43E-02	-4.75E-04
599.994	1.38E-02	1.70E-06	-1.38E-02	-4.59E-04	3.29E-05	4.26E-04	-1.48E-02	-1.96E-03	1.70E-06	-1.38E-02	-4.59E-04
609.994	1.34E-02	1.23E-06	-1.34E-02	-4.43E-04	3.13E-05	4.12E-04	-1.44E-02	-1.94E-03	1.23E-06	-1.34E-02	-4.43E-04
619.994	1.29E-02	7.63E-07	-1.29E-02	-4.27E-04	2.97E-05	3.97E-04	-1.39E-02	-1.93E-03	7.63E-07	-1.29E-02	-4.27E-04
629.994	1.25E-02	3.04E-07	-1.25E-02	-4.11E-04	2.82E-05	3.83E-04	-1.35E-02	-1.91E-03	3.04E-07	-1.25E-02	-4.11E-04
639.993	1.20E-02	-1.47E-07	-1.20E-02	-3.96E-04	2.67E-05	3.69E-04	-1.30E-02	-1.90E-03	-1.47E-07	-1.20E-02	-3.96E-04
649.993	1.16E-02	-5.88E-07	-1.16E-02	-3.81E-04	2.52E-05	3.55E-04	-1.26E-02	-1.88E-03	-5.88E-07	-1.16E-02	-3.81E-04
659.993	1.11E-02	-1.02E-06	-1.11E-02	-3.66E-04	2.37E-05	3.42E-04	-1.21E-02	-1.87E-03	-1.02E-06	-1.11E-02	-3.66E-04
669.993	1.07E-02	-1.44E-06	-1.07E-02	-3.51E-04	2.23E-05	3.29E-04	-1.17E-02	-1.85E-03	-1.44E-06	-1.07E-02	-3.51E-04
679.993	1.03E-02	-1.85E-06	-1.03E-02	-3.37E-04	2.10E-05	3.16E-04	-1.13E-02	-1.84E-03	-1.85E-06	-1.03E-02	-3.37E-04
689.993	9.88E-03	-2.24E-06	-9.88E-03	-3.23E-04	1.96E-05	3.03E-04	-1.09E-02	-1.82E-03	-2.24E-06	-9.88E-03	-3.23E-04
699.993	9.49E-03	-2.63E-06	-9.48E-03	-3.09E-04	1.83E-05	2.91E-04	-1.05E-02	-1.81E-03	-2.63E-06	-9.48E-03	-3.09E-04
709.992	9.10E-03	-3.00E-06	-9.10E-03	-2.96E-04	1.71E-05	2.79E-04	-1.01E-02	-1.80E-03	-3.00E-06	-9.10E-03	-2.96E-04
719.992	8.73E-03	-3.36E-06	-8.72E-03	-2.84E-04	1.59E-05	2.68E-04	-9.72E-03	-1.78E-03	-3.36E-06	-8.72E-03	-2.84E-04
729.992	8.37E-03	-3.71E-06	-8.36E-03	-2.71E-04	1.47E-05	2.57E-04	-9.36E-03	-1.77E-03	-3.71E-06	-8.36E-03	-2.71E-04
739.992	8.02E-03	-4.05E-06	-8.01E-03	-2.59E-04	1.35E-05	2.46E-04	-9.01E-03	-1.76E-03	-4.05E-06	-8.01E-03	-2.59E-04
749.992	7.68E-03	-4.37E-06	-7.67E-03	-2.48E-04	1.24E-05	2.35E-04	-8.67E-03	-1.75E-03	-4.37E-06	-7.67E-03	-2.48E-04
759.992	7.35E-03	-4.68E-06	-7.34E-03	-2.36E-04	1.14E-05	2.25E-04	-8.34E-03	-1.74E-03	-4.68E-06	-7.34E-03	-2.36E-04
769.991	7.03E-03	-4.98E-06	-7.02E-03	-2.26E-04	1.03E-05	2.15E-04	-8.02E-03	-1.73E-03	-4.98E-06	-7.02E-03	-2.26E-04
779.991	6.72E-03	-5.26E-06	-6.72E-03	-2.15E-04	9.36E-06	2.06E-04	-7.72E-03	-1.72E-03	-5.26E-06	-6.72E-03	-2.15E-04
789.991	6.43E-03	-5.54E-06	-6.42E-03	-2.05E-04	8.41E-06	1.97E-04	-7.42E-03	-1.71E-03	-5.54E-06	-6.42E-03	-2.05E-04
799.991	6.14E-03	-5.80E-06	-6.14E-03	-1.95E-04	7.50E-06	1.88E-04	-7.14E-03	-1.70E-03	-5.80E-06	-6.14E-03	-1.95E-04
809.991	5.87E-03	-6.05E-06	-5.86E-03	-1.86E-04	6.63E-06	1.79E-04	-6.86E-03	-1.69E-03	-6.05E-06	-5.86E-03	-1.86E-04
819.991	5.60E-03	-6.29E-06	-5.60E-03	-1.77E-04	5.79E-06	1.71E-04	-6.60E-03	-1.68E-03	-6.29E-06	-5.60E-03	-1.77E-04
829.990	5.35E-03	-6.52E-06	-5.34E-03	-1.68E-04	4.98E-06	1.63E-04	-6.34E-03	-1.67E-03	-6.52E-06	-5.34E-03	-1.68E-04
839.990	5.11E-03	-6.74E-06	-5.10E-03	-1.60E-04	4.21E-06	1.56E-04	-6.10E-03	-1.66E-03	-6.74E-06	-5.10E-03	-1.60E-04
849.990	4.87E-03	-6.94E-06	-4.87E-03	-1.52E-04	3.48E-06	1.49E-04	-5.87E-03	-1.65E-03	-6.94E-06	-4.87E-03	-1.52E-04
859.990	4.65E-03	-7.14E-06	-4.64E-03	-1.45E-04	2.77E-06	1.42E-04	-5.64E-03	-1.64E-03	-7.14E-06	-4.64E-03	-1.45E-04
869.990	4.43E-03	-7.33E-06	-4.42E-03	-1.37E-04	2.10E-06	1.35E-04	-5.42E-03	-1.64E-03	-7.33E-06	-4.42E-03	-1.37E-04
879.990	4.23E-03	-7.50E-06	-4.22E-03	-1.30E-04	1.45E-06	1.29E-04	-5.22E-03	-1.63E-03	-7.50E-06	-4.22E-03	-1.30E-04
889.990	4.03E-03	-7.67E-06	-4.02E-03	-1.24E-04	8.39E-07	1.23E-04	-5.02E-03	-1.62E-03	-7.67E-06	-4.02E-03	-1.24E-04
899.989	3.84E-03	-7.83E-06	-3.83E-03	-1.17E-04	2.52E-07	1.17E-04	-4.83E-03	-1.62E-03	-7.83E-06	-3.83E-03	-1.17E-04
909.989	3.66E-03	-7.98E-06	-3.65E-03	-1.11E-04	-3.07E-07	1.11E-04	-4.65E-03	-1.61E-03	-7.98E-06	-3.65E-03	-1.11E-04
919.989	3.48E-03	-8.12E-06	-3.47E-03	-1.05E-04	-8.41E-07	1.06E-04	-4.47E-03	-1.61E-03	-8.12E-06	-3.47E-03	-1.05E-04
929.989	3.31E-03	-8.26E-06	-3.31E-03	-9.95E-05	-1.35E-06	1.01E-04	-4.31E-03	-1.60E-03	-8.26E-06	-3.31E-03	-9.95E-05
939.989	3.16E-03	-8.38E-06	-3.15E-03	-9.41E-05	-1.83E-06	9.59E-05	-4.15E-03	-1.59E-03	-8.38E-06	-3.15E-03	-9.41E-05
949.989	3.00E-03	-8.50E-06	-2.99E-03	-8.89E-05	-2.29E-06	9.12E-05	-3.99E-03	-1.59E-03	-8.50E-06	-2.99E-03	-8.89E-05
959.988	2.86E-03	-8.61E-06	-2.85E-03	-8.40E-05	-2.73E-06	8.68E-05	-3.85E-03	-1.58E-03	-8.61E-06	-2.85E-03	-8.40E-05
969.988	2.72E-03	-8.72E-06	-2.71E-03	-7.93E-05	-3.14E-06	8.25E-05	-3.71E-03	-1.58E-03	-8.72E-06	-2.71E-03	-7.93E-05
979.988	2.59E-03	-8.81E-06	-2.58E-03	-7.49E-05	-3.54E-06	7.84E-05	-3.58E-03	-1.57E-03	-8.81E-06	-2.58E-03	-7.49E-05
989.988	2.46E-03	-8.90E-06	-2.45E-03	-7.06E-05	-3.91E-06	7.45E-05	-3.45E-03	-1.57E-03	-8.90E-06	-2.45E-03	-7.06E-05

TIME (USEC)	IL4	IC1	IC2	IC3	PJ1	PJ2	PV1	VR1	VR2	VR3	VR4
499.996	4.85E-05	6.30E-06	-1.89E-02	4.85E-05	1.00E-03	1.50E-03	0.0	0.914	0.315	-1.828	-6.15E-02
509.996	4.71E-05	5.89E-06	-1.85E-02	4.71E-05	1.00E-03	1.50E-03	0.0	0.895	0.295	-1.790	-6.01E-02
519.995	4.56E-05	5.46E-06	-1.81E-02	4.56E-05	1.00E-03	1.50E-03	0.0	0.874	0.273	-1.749	-5.87E-02
529.995	4.41E-05	5.02E-06	-1.76E-02	4.41E-05	1.00E-03	1.50E-03	0.0	0.853	0.251	-1.707	-5.72E-02
539.995	4.25E-05	4.56E-06	-1.72E-02	4.25E-05	1.00E-03	1.50E-03	0.0	0.831	0.228	-1.663	-5.56E-02
549.995	4.09E-05	4.09E-06	-1.67E-02	4.09E-05	1.00E-03	1.50E-03	0.0	0.809	0.205	-1.618	-5.40E-02
559.995	3.93E-05	3.62E-06	-1.62E-02	3.93E-05	1.00E-03	1.50E-03	0.0	0.786	0.181	-1.572	-5.24E-02
569.995	3.77E-05	3.14E-06	-1.58E-02	3.77E-05	1.00E-03	1.50E-03	0.0	0.762	0.157	-1.525	-5.08E-02
579.994	3.61E-05	2.66E-06	-1.53E-02	3.61E-05	1.00E-03	1.50E-03	0.0	0.739	0.133	-1.478	-4.92E-02
589.994	3.45E-05	2.18E-06	-1.48E-02	3.45E-05	1.00E-03	1.50E-03	0.0	0.715	0.109	-1.431	-4.75E-02
599.994	3.29E-05	1.70E-06	-1.43E-02	3.29E-05	1.00E-03	1.50E-03	0.0	0.692	8.51E-02	-1.384	-4.59E-02
609.994	3.13E-05	1.23E-06	-1.38E-02	3.13E-05	1.00E-03	1.50E-03	0.0	0.669	6.14E-02	-1.337	-4.43E-02
619.994	2.97E-05	7.63E-07	-1.33E-02	2.97E-05	1.00E-03	1.50E-03	0.0	0.645	3.81E-02	-1.291	-4.27E-02
629.994	2.82E-05	3.04E-07	-1.29E-02	2.82E-05	1.00E-03	1.50E-03	0.0	0.623	1.52E-02	-1.245	-4.11E-02
639.993	2.67E-05	-1.47E-07	-1.24E-02	2.67E-05	1.00E-03	1.50E-03	0.0	0.600	-7.34E-03	-1.200	-3.96E-02
649.993	2.52E-05	-5.88E-07	-1.19E-02	2.52E-05	1.00E-03	1.50E-03	0.0	0.578	-2.94E-02	-1.156	-3.81E-02
659.993	2.37E-05	-1.02E-06	-1.15E-02	2.37E-05	1.00E-03	1.50E-03	0.0	0.556	-5.09E-02	-1.112	-3.66E-02
669.993	2.23E-05	-1.44E-06	-1.11E-02	2.23E-05	1.00E-03	1.50E-03	0.0	0.535	-7.19E-02	-1.070	-3.51E-02
679.993	2.10E-05	-1.85E-06	-1.06E-02	2.10E-05	1.00E-03	1.50E-03	0.0	0.514	-9.24E-02	-1.028	-3.37E-02
689.993	1.96E-05	-2.24E-06	-1.02E-02	1.96E-05	1.00E-03	1.50E-03	0.0	0.494	-0.112	-0.988	-3.23E-02
699.993	1.83E-05	-2.63E-06	-9.79E-03	1.83E-05	1.00E-03	1.50E-03	0.0	0.474	-0.131	-0.948	-3.09E-02
709.992	1.71E-05	-3.00E-06	-9.39E-03	1.71E-05	1.00E-03	1.50E-03	0.0	0.455	-0.150	-0.910	-2.96E-02
719.992	1.59E-05	-3.36E-06	-9.01E-03	1.59E-05	1.00E-03	1.50E-03	0.0	0.436	-0.168	-0.872	-2.84E-02
729.992	1.47E-05	-3.71E-06	-8.63E-03	1.47E-05	1.00E-03	1.50E-03	0.0	0.418	-0.186	-0.836	-2.71E-02
739.992	1.35E-05	-4.05E-06	-8.27E-03	1.35E-05	1.00E-03	1.50E-03	0.0	0.401	-0.202	-0.801	-2.59E-02
749.992	1.24E-05	-4.37E-06	-7.92E-03	1.24E-05	1.00E-03	1.50E-03	0.0	0.384	-0.218	-0.767	-2.48E-02
759.992	1.14E-05	-4.68E-06	-7.58E-03	1.14E-05	1.00E-03	1.50E-03	0.0	0.367	-0.234	-0.734	-2.36E-02
769.991	1.03E-05	-4.98E-06	-7.25E-03	1.03E-05	1.00E-03	1.50E-03	0.0	0.351	-0.249	-0.702	-2.26E-02
779.991	9.36E-06	-5.26E-06	-6.93E-03	9.36E-06	1.00E-03	1.50E-03	0.0	0.336	-0.263	-0.672	-2.15E-02
789.991	8.41E-06	-5.54E-06	-6.63E-03	8.41E-06	1.00E-03	1.50E-03	0.0	0.321	-0.277	-0.642	-2.05E-02
799.991	7.50E-06	-5.80E-06	-6.33E-03	7.50E-06	1.00E-03	1.50E-03	0.0	0.307	-0.290	-0.614	-1.95E-02
809.991	6.63E-06	-6.05E-06	-6.05E-03	6.63E-06	1.00E-03	1.50E-03	0.0	0.293	-0.303	-0.586	-1.86E-02
819.991	5.79E-06	-6.29E-06	-5.78E-03	5.79E-06	1.00E-03	1.50E-03	0.0	0.280	-0.314	-0.560	-1.77E-02
829.990	4.98E-06	-6.52E-06	-5.51E-03	4.98E-06	1.00E-03	1.50E-03	0.0	0.268	-0.326	-0.534	-1.68E-02
839.990	4.21E-06	-6.74E-06	-5.26E-03	4.21E-06	1.00E-03	1.50E-03	0.0	0.255	-0.337	-0.510	-1.60E-02
849.990	3.48E-06	-6.94E-06	-5.02E-03	3.48E-06	1.00E-03	1.50E-03	0.0	0.244	-0.347	-0.487	-1.52E-02
859.990	2.77E-06	-7.14E-06	-4.79E-03	2.77E-06	1.00E-03	1.50E-03	0.0	0.232	-0.357	-0.464	-1.45E-02
869.990	2.10E-06	-7.33E-06	-4.56E-03	2.10E-06	1.00E-03	1.50E-03	0.0	0.222	-0.366	-0.442	-1.37E-02
879.990	1.45E-06	-7.50E-06	-4.35E-03	1.45E-06	1.00E-03	1.50E-03	0.0	0.211	-0.375	-0.422	-1.30E-02
889.990	8.39E-07	-7.67E-06	-4.14E-03	8.39E-07	1.00E-03	1.50E-03	0.0	0.201	-0.384	-0.402	-1.24E-02
899.989	2.52E-07	-7.83E-06	-3.95E-03	2.52E-07	1.00E-03	1.50E-03	0.0	0.192	-0.391	-0.383	-1.17E-02
909.989	-3.07E-07	-7.98E-06	-3.76E-03	-3.07E-07	1.00E-03	1.50E-03	0.0	0.183	-0.399	-0.365	-1.11E-02
919.989	-8.41E-07	-8.12E-06	-3.58E-03	-8.41E-07	1.00E-03	1.50E-03	0.0	0.174	-0.406	-0.347	-1.05E-02
929.989	-1.35E-06	-8.26E-06	-3.41E-03	-1.35E-06	1.00E-03	1.50E-03	0.0	0.166	-0.413	-0.331	-9.95E-03
939.989	-1.83E-06	-8.38E-06	-3.24E-03	-1.83E-06	1.00E-03	1.50E-03	0.0	0.158	-0.419	-0.315	-9.41E-03
949.989	-2.29E-06	-8.50E-06	-3.08E-03	-2.29E-06	1.00E-03	1.50E-03	0.0	0.150	-0.425	-0.299	-8.89E-03
959.988	-2.73E-06	-8.61E-06	-2.93E-03	-2.73E-06	1.00E-03	1.50E-03	0.0	0.143	-0.431	-0.285	-8.40E-03
969.988	-3.14E-06	-8.72E-06	-2.79E-03	-3.14E-06	1.00E-03	1.50E-03	0.0	0.136	-0.436	-0.271	-7.93E-03
979.988	-3.54E-06	-8.81E-06	-2.65E-03	-3.54E-06	1.00E-03	1.50E-03	0.0	0.129	-0.441	-0.258	-7.49E-03
989.988	-3.91E-06	-8.90E-06	-2.52E-03	-3.91E-06	1.00E-03	1.50E-03	0.0	0.123	-0.445	-0.245	-7.06E-03

TIME (USEC)	VR5	VR6	VR7	VR8	VL1	VL2	VL3	VL4	VC1	VC2	VC3
499.996	2.423	2.830	-0.193	-2.11E-02	-3.91E-05	0.455	0.434	-1.37E-04	0.599	0.479	0.407
509.996	2.353	2.769	-0.189	-2.10E-02	-4.16E-05	0.488	0.465	-1.44E-04	0.600	0.385	0.417
519.995	2.279	2.705	-0.185	-2.09E-02	-4.36E-05	0.515	0.491	-1.50E-04	0.601	0.294	0.426
529.995	2.203	2.638	-0.181	-2.07E-02	-4.52E-05	0.536	0.511	-1.54E-04	0.602	0.204	0.435
539.995	2.125	2.568	-0.176	-2.06E-02	-4.63E-05	0.553	0.527	-1.58E-04	0.603	0.117	0.443
549.995	2.045	2.497	-0.172	-2.04E-02	-4.72E-05	0.566	0.539	-1.59E-04	0.604	3.24E-02	0.452
559.995	1.965	2.425	-0.167	-2.02E-02	-4.77E-05	0.574	0.548	-1.61E-04	0.605	-5.01E-02	0.460
569.995	1.884	2.352	-0.162	-2.01E-02	-4.79E-05	0.580	0.553	-1.61E-04	0.605	-0.130	0.468
579.994	1.804	2.278	-0.158	-1.99E-02	-4.79E-05	0.582	0.555	-1.62E-04	0.606	-0.208	0.475
589.994	1.723	2.205	-0.153	-1.98E-02	-4.78E-05	0.582	0.555	-1.61E-04	0.607	-0.283	0.482
599.994	1.643	2.132	-0.148	-1.96E-02	-4.74E-05	0.580	0.553	-1.59E-04	0.607	-0.356	0.489
609.994	1.564	2.059	-0.144	-1.94E-02	-4.69E-05	0.576	0.549	-1.57E-04	0.607	-0.426	0.495
619.994	1.486	1.987	-0.139	-1.93E-02	-4.63E-05	0.570	0.543	-1.56E-04	0.607	-0.494	0.501
629.994	1.409	1.916	-0.135	-1.91E-02	-4.54E-05	0.562	0.536	-1.52E-04	0.607	-0.560	0.507
639.993	1.334	1.846	-0.130	-1.90E-02	-4.46E-05	0.553	0.527	-1.49E-04	0.608	-0.623	0.513
649.993	1.260	1.777	-0.126	-1.88E-02	-4.36E-05	0.543	0.518	-1.47E-04	0.607	-0.684	0.518
659.993	1.187	1.710	-0.121	-1.87E-02	-4.26E-05	0.532	0.508	-1.42E-04	0.607	-0.742	0.523
669.993	1.117	1.644	-0.117	-1.85E-02	-4.14E-05	0.521	0.496	-1.39E-04	0.607	-0.799	0.527
679.993	1.048	1.580	-0.113	-1.84E-02	-4.03E-05	0.509	0.485	-1.35E-04	0.607	-0.853	0.532
689.993	0.981	1.517	-0.109	-1.82E-02	-3.91E-05	0.496	0.473	-1.32E-04	0.606	-0.905	0.536
699.993	0.916	1.456	-0.105	-1.81E-02	-3.78E-05	0.483	0.460	-1.28E-04	0.606	-0.955	0.539
709.992	0.854	1.396	-0.101	-1.80E-02	-3.66E-05	0.469	0.447	-1.24E-04	0.605	-1.003	0.543
719.992	0.793	1.339	-9.72E-02	-1.78E-02	-3.54E-05	0.455	0.434	-1.20E-04	0.605	-1.049	0.546
729.992	0.734	1.283	-9.36E-02	-1.77E-02	-3.42E-05	0.442	0.421	-1.16E-04	0.604	-1.093	0.549
739.992	0.676	1.228	-9.01E-02	-1.76E-02	-3.29E-05	0.428	0.408	-1.12E-04	0.603	-1.136	0.552
749.992	0.621	1.176	-8.67E-02	-1.75E-02	-3.17E-05	0.414	0.394	-1.08E-04	0.602	-1.176	0.555
759.992	0.568	1.125	-8.34E-02	-1.74E-02	-3.04E-05	0.400	0.381	-1.05E-04	0.601	-1.215	0.557
769.991	0.517	1.076	-8.02E-02	-1.73E-02	-2.92E-05	0.386	0.368	-1.01E-04	0.600	-1.252	0.559
779.991	0.468	1.029	-7.72E-02	-1.72E-02	-2.80E-05	0.373	0.355	-9.69E-05	0.599	-1.288	0.561
789.991	0.421	0.983	-7.42E-02	-1.71E-02	-2.68E-05	0.359	0.343	-9.27E-05	0.598	-1.322	0.563
799.991	0.375	0.940	-7.14E-02	-1.70E-02	-2.56E-05	0.346	0.330	-8.91E-05	0.597	-1.354	0.565
809.991	0.331	0.897	-6.86E-02	-1.69E-02	-2.45E-05	0.333	0.318	-8.55E-05	0.596	-1.385	0.566
819.991	0.289	0.857	-6.60E-02	-1.68E-02	-2.34E-05	0.321	0.306	-8.20E-05	0.595	-1.415	0.567
829.990	0.249	0.817	-6.34E-02	-1.67E-02	-2.22E-05	0.308	0.294	-7.86E-05	0.593	-1.443	0.568
839.990	0.211	0.780	-6.10E-02	-1.66E-02	-2.13E-05	0.296	0.282	-7.53E-05	0.592	-1.470	0.569
849.990	0.174	0.744	-5.87E-02	-1.65E-02	-2.01E-05	0.284	0.271	-7.20E-05	0.591	-1.495	0.570
859.990	0.139	0.709	-5.64E-02	-1.64E-02	-1.91E-05	0.273	0.260	-6.88E-05	0.589	-1.520	0.571
869.990	0.105	0.676	-5.42E-02	-1.64E-02	-1.82E-05	0.262	0.249	-6.58E-05	0.588	-1.543	0.571
879.990	7.27E-02	0.644	-5.22E-02	-1.63E-02	-1.73E-05	0.251	0.239	-6.28E-05	0.586	-1.566	0.572
889.990	4.19E-02	0.614	-5.02E-02	-1.62E-02	-1.64E-05	0.240	0.229	-6.00E-05	0.585	-1.587	0.572
899.989	1.26E-02	0.584	-4.83E-02	-1.62E-02	-1.54E-05	0.230	0.219	-5.73E-05	0.583	-1.607	0.572
909.989	-1.54E-02	0.556	-4.65E-02	-1.61E-02	-1.46E-05	0.220	0.210	-5.47E-05	0.582	-1.626	0.572
919.989	-4.20E-02	0.530	-4.47E-02	-1.61E-02	-1.38E-05	0.211	0.201	-5.20E-05	0.580	-1.645	0.572
929.989	-6.74E-02	0.504	-4.31E-02	-1.60E-02	-1.30E-05	0.202	0.192	-4.95E-05	0.579	-1.662	0.572
939.989	-9.16E-02	0.480	-4.15E-02	-1.59E-02	-1.23E-05	0.193	0.184	-4.71E-05	0.577	-1.679	0.571
949.989	-0.115	0.456	-3.99E-02	-1.59E-02	-1.15E-05	0.184	0.176	-4.48E-05	0.575	-1.695	0.571
959.988	-0.136	0.434	-3.85E-02	-1.58E-02	-1.08E-05	0.176	0.168	-4.26E-05	0.573	-1.710	0.570
969.988	-0.157	0.412	-3.71E-02	-1.58E-02	-1.01E-05	0.168	0.160	-4.05E-05	0.572	-1.724	0.570
979.988	-0.177	0.392	-3.58E-02	-1.57E-02	-9.42E-06	0.160	0.153	-3.84E-05	0.570	-1.738	0.569
989.988	-0.196	0.373	-3.45E-02	-1.57E-02	-8.82E-06	0.153	0.146	-3.64E-05	0.568	-1.751	0.568

TIME (USEC)	VPJ1	VPJ2
499.996	-0.193	-2.11E-02
509.996	-0.189	-2.10E-02
519.995	-0.185	-2.09E-02
529.995	-0.181	-2.07E-02
539.995	-0.176	-2.06E-02
549.995	-0.172	-2.04E-02
559.995	-0.167	-2.02E-02
569.995	-0.162	-2.01E-02
579.994	-0.158	-1.99E-02
589.994	-0.153	-1.98E-02
599.994	-0.148	-1.96E-02
609.994	-0.144	-1.94E-02
619.994	-0.139	-1.93E-02
629.994	-0.135	-1.91E-02
639.993	-0.130	-1.90E-02
649.993	-0.126	-1.88E-02
659.993	-0.121	-1.87E-02
669.993	-0.117	-1.85E-02
679.993	-0.113	-1.84E-02
689.993	-0.109	-1.82E-02
699.993	-0.105	-1.81E-02
709.992	-0.101	-1.80E-02
719.992	-9.72E-02	-1.78E-02
729.992	-9.36E-02	-1.77E-02
739.992	-9.01E-02	-1.76E-02
749.992	-8.67E-02	-1.75E-02
759.992	-8.34E-02	-1.74E-02
769.991	-8.02E-02	-1.73E-02
779.991	-7.72E-02	-1.72E-02
789.991	-7.42E-02	-1.71E-02
799.991	-7.14E-02	-1.70E-02
809.991	-6.86E-02	-1.69E-02
819.991	-6.60E-02	-1.68E-02
829.990	-6.34E-02	-1.67E-02
839.990	-6.10E-02	-1.66E-02
849.990	-5.87E-02	-1.65E-02
859.990	-5.64E-02	-1.64E-02
869.990	-5.42E-02	-1.64E-02
879.990	-5.22E-02	-1.63E-02
889.990	-5.02E-02	-1.62E-02
899.989	-4.83E-02	-1.62E-02
909.989	-4.65E-02	-1.61E-02
919.989	-4.47E-02	-1.61E-02
929.989	-4.31E-02	-1.60E-02
939.989	-4.15E-02	-1.59E-02
949.989	-3.99E-02	-1.59E-02
959.988	-3.85E-02	-1.58E-02
969.988	-3.71E-02	-1.58E-02
979.988	-3.58E-02	-1.57E-02
989.988	-3.45E-02	-1.57E-02

END L5 9999947

03 JUN 70

TIME	IR1	IR2	IR3	IR4	IR5	IR6	IR7	IR8	IL1	IL2	IL3
(USEC)											
999.988	2.34E-03	-8.99E-06	-2.33E-03	-6.65E-05	-4.27E-06	7.08E-05	-3.33E-03	-1.57E-03	-8.99E-06	-2.33E-03	-6.65E-05
(MSEC)											
1.010	-7.19E-03	1.81E-04	7.01E-03	-1.81E-03	1.53E-04	1.65E-03	6.01E-03	-3.31E-03	1.81E-04	7.01E-03	-1.81E-03
1.020	-1.37E-02	1.74E-04	1.36E-02	-1.59E-03	1.33E-04	1.46E-03	1.26E-02	-3.09E-03	1.74E-04	1.36E-02	-1.59E-03
1.030	-1.94E-02	1.67E-04	1.93E-02	-1.40E-03	1.16E-04	1.29E-03	1.83E-02	-2.90E-03	1.67E-04	1.93E-02	-1.40E-03
1.040	-2.44E-02	1.62E-04	2.42E-02	-1.24E-03	1.01E-04	1.14E-03	2.32E-02	-2.74E-03	1.62E-04	2.42E-02	-1.24E-03
1.050	-2.86E-02	1.57E-04	2.84E-02	-1.11E-03	8.79E-05	1.02E-03	2.74E-02	-2.61E-03	1.57E-04	2.84E-02	-1.11E-03
1.060	-3.22E-02	1.52E-04	3.21E-02	-9.87E-04	7.69E-05	9.11E-04	3.11E-02	-2.49E-03	1.52E-04	3.21E-02	-9.87E-04
1.070	-3.53E-02	1.49E-04	3.52E-02	-8.88E-04	6.76E-05	8.20E-04	3.42E-02	-2.39E-03	1.49E-04	3.52E-02	-8.88E-04
1.080	-3.79E-02	1.46E-04	3.77E-02	-8.05E-04	5.98E-05	7.46E-04	3.67E-02	-2.31E-03	1.46E-04	3.77E-02	-8.05E-04
1.090	-4.00E-02	1.43E-04	3.98E-02	-7.38E-04	5.35E-05	6.84E-04	3.88E-02	-2.24E-03	1.43E-04	3.98E-02	-7.38E-04
1.100	-4.17E-02	1.41E-04	4.16E-02	-6.83E-04	4.83E-05	6.35E-04	4.06E-02	-2.18E-03	1.41E-04	4.16E-02	-6.83E-04
1.110	-4.30E-02	1.39E-04	4.29E-02	-6.41E-04	4.43E-05	5.97E-04	4.19E-02	-2.14E-03	1.39E-04	4.29E-02	-6.41E-04
1.120	-4.41E-02	1.37E-04	4.39E-02	-6.09E-04	4.13E-05	5.68E-04	4.29E-02	-2.11E-03	1.37E-04	4.39E-02	-6.09E-04
1.130	-4.48E-02	1.36E-04	4.47E-02	-5.87E-04	3.91E-05	5.48E-04	4.37E-02	-2.09E-03	1.36E-04	4.47E-02	-5.87E-04
1.140	-4.53E-02	1.35E-04	4.52E-02	-5.74E-04	3.78E-05	5.36E-04	4.42E-02	-2.07E-03	1.35E-04	4.52E-02	-5.74E-04
1.150	-4.55E-02	1.34E-04	4.54E-02	-5.68E-04	3.71E-05	5.30E-04	4.44E-02	-2.07E-03	1.34E-04	4.54E-02	-5.68E-04
1.160	-4.56E-02	1.33E-04	4.55E-02	-5.68E-04	3.70E-05	5.31E-04	4.45E-02	-2.07E-03	1.33E-04	4.55E-02	-5.68E-04
1.170	-4.55E-02	1.33E-04	4.53E-02	-5.74E-04	3.74E-05	5.37E-04	4.43E-02	-2.07E-03	1.33E-04	4.53E-02	-5.74E-04
1.180	-4.52E-02	1.33E-04	4.51E-02	-5.85E-04	3.83E-05	5.47E-04	4.41E-02	-2.09E-03	1.33E-04	4.51E-02	-5.85E-04
1.190	-4.48E-02	1.33E-04	4.46E-02	-6.01E-04	3.96E-05	5.62E-04	4.36E-02	-2.10E-03	1.33E-04	4.46E-02	-6.01E-04
1.200	-4.42E-02	1.33E-04	4.41E-02	-6.21E-04	4.12E-05	5.79E-04	4.31E-02	-2.12E-03	1.33E-04	4.41E-02	-6.21E-04

TIME	IL4	IC1	IC2	IC3	PJ1	PJ2	PV1	VR1	VR2	VR3	VR4
(USEC)											
999.988	-4.27E-06	-8.99E-06	-2.40E-03	-4.27E-06	1.00E-03	1.50E-03	0.0	0.117	-0.449	-0.233	-6.65E-03
(MSEC)											
1.010	1.53E-04	1.81E-04	5.21E-03	1.53E-04	1.00E-03	1.50E-03	10.00	-0.760	9.038	0.701	-0.181
1.020	1.33E-04	1.74E-04	1.20E-02	1.33E-04	1.00E-03	1.50E-03	10.00	-0.686	8.676	1.355	-0.159
1.030	1.16E-04	1.67E-04	1.78E-02	1.16E-04	1.00E-03	1.50E-03	10.00	-0.971	8.357	1.925	-0.140
1.040	1.01E-04	1.62E-04	2.29E-02	1.01E-04	1.00E-03	1.50E-03	10.00	-1.218	8.077	2.419	-0.124
1.050	8.79E-05	1.57E-04	2.73E-02	8.79E-05	1.00E-03	1.50E-03	10.00	-1.430	7.833	2.844	-0.111
1.060	7.69E-05	1.52E-04	3.11E-02	7.69E-05	1.00E-03	1.50E-03	10.00	-1.611	7.620	3.208	-9.87E-02
1.070	6.76E-05	1.49E-04	3.43E-02	6.76E-05	1.00E-03	1.50E-03	10.00	-1.765	7.437	3.515	-8.88E-02
1.080	5.98E-05	1.46E-04	3.69E-02	5.98E-05	1.00E-03	1.50E-03	10.00	-1.893	7.279	3.772	-8.05E-02
1.090	5.35E-05	1.43E-04	3.91E-02	5.35E-05	1.00E-03	1.50E-03	10.00	-1.999	7.144	3.984	-7.38E-02
1.100	4.83E-05	1.41E-04	4.09E-02	4.83E-05	1.00E-03	1.50E-03	10.00	-2.085	7.030	4.156	-6.83E-02
1.110	4.43E-05	1.39E-04	4.23E-02	4.43E-05	1.00E-03	1.50E-03	10.00	-2.152	6.935	4.291	-6.41E-02
1.120	4.13E-05	1.37E-04	4.33E-02	4.13E-05	1.00E-03	1.50E-03	10.00	-2.204	6.856	4.393	-6.09E-02
1.130	3.91E-05	1.36E-04	4.41E-02	3.91E-05	1.00E-03	1.50E-03	10.00	-2.240	6.792	4.467	-5.87E-02
1.140	3.78E-05	1.35E-04	4.46E-02	3.78E-05	1.00E-03	1.50E-03	10.00	-2.264	6.741	4.515	-5.74E-02
1.150	3.71E-05	1.34E-04	4.48E-02	3.71E-05	1.00E-03	1.50E-03	10.00	-2.277	6.701	4.540	-5.68E-02
1.160	3.70E-05	1.33E-04	4.49E-02	3.70E-05	1.00E-03	1.50E-03	10.00	-2.279	6.672	4.546	-5.68E-02
1.170	3.74E-05	1.33E-04	4.48E-02	3.74E-05	1.00E-03	1.50E-03	10.00	-2.273	6.651	4.533	-5.74E-02
1.180	3.83E-05	1.33E-04	4.45E-02	3.83E-05	1.00E-03	1.50E-03	10.00	-2.259	6.639	4.505	-5.85E-02
1.190	3.96E-05	1.33E-04	4.40E-02	3.96E-05	1.00E-03	1.50E-03	10.00	-2.239	6.633	4.464	-6.01E-02
1.200	4.12E-05	1.33E-04	4.35E-02	4.12E-05	1.00E-03	1.50E-03	10.00	-2.212	6.633	4.411	-6.21E-02

TIME	VR5	VR6	VR7	VR8	VL1	VL2	VL3	VL4	VC1	VC2	VC3
(USEC)											
999.988	-0.213	0.354	-3.33E-02	-1.57E-02	-8.23E-06	0.146	0.139	-3.45E-05	0.566	-1.763	0.567
(MSEC)											
1.010	7.663	8.260	6.01E-02	-3.31E-02	-7.69E-04	8.634	8.229	-2.14E-03	0.603	-1.756	0.599
1.020	6.663	7.289	0.126	-3.09E-02	-6.78E-04	7.545	7.191	-1.86E-03	0.639	-1.712	0.628
1.030	5.793	6.444	0.183	-2.90E-02	-5.96E-04	6.559	6.251	-1.62E-03	0.673	-1.637	0.653
1.040	5.041	5.714	0.232	-2.74E-02	-5.21E-04	5.667	5.401	-1.39E-03	0.706	-1.535	0.675
1.050	4.394	5.087	0.274	-2.61E-02	-4.55E-04	4.860	4.632	-1.19E-03	0.738	-1.409	0.693
1.060	3.844	4.553	0.311	-2.49E-02	-3.94E-04	4.133	3.939	-1.01E-03	0.768	-1.262	0.710
1.070	3.379	4.102	0.342	-2.39E-02	-3.40E-04	3.477	3.314	-8.48E-04	0.799	-1.099	0.724
1.080	2.992	3.728	0.367	-2.31E-02	-2.91E-04	2.887	2.752	-7.02E-04	0.828	-0.920	0.737
1.090	2.673	3.421	0.388	-2.24E-02	-2.47E-04	2.358	2.247	-5.71E-04	0.857	-0.730	0.748
1.100	2.417	3.175	0.406	-2.18E-02	-2.07E-04	1.884	1.795	-4.55E-04	0.885	-0.530	0.759
1.110	2.216	2.984	0.419	-2.14E-02	-1.72E-04	1.459	1.391	-3.51E-04	0.913	-0.321	0.768
1.120	2.065	2.841	0.429	-2.11E-02	-1.41E-04	1.081	1.030	-2.58E-04	0.941	-0.107	0.776
1.130	1.957	2.741	0.437	-2.09E-02	-1.13E-04	0.744	0.709	-1.75E-04	0.968	0.112	0.784
1.140	1.888	2.680	0.442	-2.07E-02	-8.95E-05	0.445	0.424	-1.03E-04	0.995	0.334	0.792
1.150	1.853	2.652	0.444	-2.07E-02	-6.68E-05	0.181	0.172	-3.82E-05	1.022	0.558	0.800
1.160	1.848	2.655	0.445	-2.07E-02	-4.86E-05	-5.20E-02	-4.98E-02	1.79E-05	1.049	0.782	0.807
1.170	1.869	2.684	0.443	-2.07E-02	-3.15E-05	-0.256	-0.245	6.75E-05	1.076	1.007	0.814
1.180	1.914	2.736	0.441	-2.09E-02	-1.72E-05	-0.435	-0.415	1.10E-04	1.102	1.230	0.822
1.190	1.978	2.808	0.436	-2.10E-02	-4.77E-06	-0.590	-0.563	1.46E-04	1.129	1.452	0.830
1.200	2.059	2.897	0.431	-2.12E-02	6.68E-06	-0.724	-0.691	1.77E-04	1.155	1.671	0.838

TIME	VPJ1	VPJ2
(USEC)		
999.988	-3.33E-02	-1.57E-02
(MSEC)		
1.010	6.01E-02	-3.31E-02
1.020	0.126	-3.09E-02
1.030	0.183	-2.90E-02
1.040	0.232	-2.74E-02
1.050	0.274	-2.61E-02
1.060	0.311	-2.49E-02
1.070	0.342	-2.39E-02
1.080	0.367	-2.31E-02
1.090	0.388	-2.24E-02
1.100	0.406	-2.18E-02
1.110	0.419	-2.14E-02
1.120	0.429	-2.11E-02
1.130	0.437	-2.09E-02
1.140	0.442	-2.07E-02
1.150	0.444	-2.07E-02
1.160	0.445	-2.07E-02
1.170	0.443	-2.07E-02
1.180	0.441	-2.09E-02
1.190	0.436	-2.10E-02
1.200	0.431	-2.12E-02
ENDJOB	9999946	